

การพัฒนาเครื่องมือสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ

นายนิรุต มิ่งโมรา

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร

มหาบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

มหาวิทยาลัยเทคโนโลยีสุรนารี

ปีการศึกษา 2549

**DEVELOPMENT OF TEST CASE GENERATION TOOL
FOR ACCEPTANCE TESTING**

Naris Mingmora

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Engineering in Computer

Engineering

Suranaree University of Technology

Academic Year 2006

การพัฒนาเครื่องมือสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ

มหาวิทยาลัยเทคโนโลยีสุรนารี
การศึกษาตามหลักสูตรปริญญาโทบริหารธุรกิจ

อนุมัติให้บัณฑิตวิทยาลัยเป็นส่วนหนึ่งของ

คณะกรรมการสอบวิทยานิพนธ์

(รศ. ดร.กิตติศักดิ์ เกิดประสพ)

ประธานกรรมการ

(ผศ. ดร.พิชโยทัย มัทธนาภิวัฒน์)

กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)

(ผศ. ดร.เคชา ชาญศิลป์)

กรรมการ

(รศ. ดร.นิตยา เกิดประสพ)

กรรมการ

(ศ. ดร.ไพโรจน์ สัตยธรรม)

รองอธิการบดีฝ่ายวิชาการ

(รศ. น.อ. ดร.วรพจน์ จำพิศ)

คณบดีสำนักวิชาวิศวกรรมศาสตร์

นริศ มิ่งโมรา : การพัฒนาเครื่องมือสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ
(DEVELOPMENT OF TEST CASE GENERATION TOOL FOR ACCEPTANCE
TESTING) อาจารย์ที่ปรึกษา : ผศ. ดร.พิชโยทัย มหัทธนาภิวัฒน์, 162 หน้า.

ในการพัฒนาซอฟต์แวร์ ขั้นตอนหนึ่งที่สำคัญเพื่อให้ซอฟต์แวร์ที่พัฒนาขึ้นนั้น มีประสิทธิภาพน่าเชื่อถือ และถูกต้องตามความต้องการของผู้ใช้งานคือ ขั้นตอนการทดสอบ ซึ่งขั้นตอนสุดท้ายของการทดสอบก่อนการส่งมอบงานอย่างเป็นทางการให้กับผู้ใช้งานคือ การทดสอบการยอมรับ โดยจะมีวิธีการทดสอบแบบแบล็กบ็อกซ์ เพื่อให้เป็นไปตามความต้องการทางธุรกิจที่ได้ระบุไว้ในข้อกำหนดความต้องการซอฟต์แวร์ โดยที่ผู้ใช้หรือกระบวนการต่างๆ นั้นจะทำตามระบบงานที่ได้ออกแบบไว้ สำหรับการทดสอบการยอมรับ สิ่งที่เป็นสำหรับผู้ทดสอบเพื่อใช้ในการทดสอบคือ กรณีทดสอบ ซึ่งการออกแบบกรณีทดสอบเป็นเรื่องที่ยุ่งยากและใช้เวลามากสำหรับผู้ทดสอบ

ดังนั้นงานวิจัยนี้จึงมีแนวคิดในการออกแบบและพัฒนาเครื่องมือช่วยในการสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ เพื่อนำมาช่วยผู้ทดสอบในการสร้างกรณีทดสอบให้ครอบคลุมเนื้อหาที่ระบุไว้ในข้อกำหนดความต้องการซอฟต์แวร์ และนอกจากนั้น ยังช่วยในการจัดเก็บข้อมูลกรณีทดสอบที่ใช้ในการทดสอบลงฐานข้อมูล เพื่อประโยชน์ในการนำกลับมาใช้ใหม่ และเป็นการลดภาระของผู้ทดสอบอีกด้วย โดยในงานวิจัยนี้จะศึกษาถึงวิธีการในการจัดเก็บความต้องการเชิงหน้าที่ ที่ได้มาจากข้อกำหนดความต้องการซอฟต์แวร์ซึ่งอ้างอิงมาจากมาตรฐานต่างประเทศ 4 มาตรฐาน ได้แก่ มาตรฐาน IEEE Std.830-1993 มาตรฐาน ESA PSS-05-0 มาตรฐาน NASA-DID-P200 และมาตรฐาน DOD-STD-498 และนำเอาความต้องการเชิงหน้าที่ ที่ได้มาจากข้อกำหนดความต้องการซอฟต์แวร์ มาสร้างเป็นกรณีทดสอบในส่วนของการทดสอบการยอมรับ โดยกรณีทดสอบที่ได้จะครอบคลุมและตรงตามความต้องการของผู้ใช้งานมากขึ้น

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ปีการศึกษา 2549

ลายมือชื่อนักศึกษา_____

ลายมือชื่ออาจารย์ที่ปรึกษา_____

ลายมือชื่ออาจารย์ที่ปรึกษาร่วม_____

NARIS MINGMORA : DEVELOPMENT OF TEST CASE GENERATION
TOOL FOR ACCEPTANCE TESTING. THESIS ADVISOR : ASST. PROF.
PICHAYOTAI MAHATTHANAPIWAT, Ph.D., 162 PP.

ACCEPTANCE TESTING/TEST CASE/SOFTWARE REQUIREMENT
SPECIFICATION

One of the most important aspects of software development to make it reliable and conform to user requirement is testing. The last phase of testing is acceptance testing that uses black box technique to test the requirements specified in the software requirement specification. Normally, the design of test cases for the acceptance testing used considerable time to consider.

This research designed and developed tool of test case generation for acceptance testing to help software testers in the testing phase. Furthermore, a database was created to record the test cases for the reusability. The research studied the methodology to store the functional requirement, in four standards: IEEE Std.830-1993 standard, ESA PSS-05-0 standard, NASA-DID-P200 standard and DOD-STD-498standard. The functional requirement from the software requirement specification will be used to generate test cases for the acceptance testing. The result produces test cases that can cover the requirement of the end user.

School of Computer Engineering.

Academic Year 2006

Student's Signature_____

Advisor's Signature_____

Co-advisor's Signature_____

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงด้วยดี ผู้วิจัยขอกราบขอบพระคุณ บุคคลต่าง ๆ ที่ได้กรุณาให้คำปรึกษา แนะนำ ช่วยเหลือ ทั้งในด้านวิชาการ และการดำเนินงานวิจัยดังต่อไปนี้

- ผู้ช่วยศาสตราจารย์ ดร.พิชโยทัย มหัทธนาภิวัฒน์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่กรุณาให้คำแนะนำในการตรวจสอบวิทยานิพนธ์ฉบับนี้
- ผู้ช่วยศาสตราจารย์ ดร.กะชา ชาญศิลป์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม
- รองศาสตราจารย์ ดร.กิตติศักดิ์ เกิดประสพ
- รองศาสตราจารย์ ดร.นิตยา เกิดประสพ
- เพื่อน ๆ บัณฑิตศึกษาทุกท่าน ที่ให้คำปรึกษา กำลังใจ และให้ความช่วยเหลืออย่างดีมาโดยตลอด
- คุณนฤมล พร้อมสันเทียะ พี่สาวที่ได้สนับสนุนทุนการศึกษา และคอยให้กำลังใจแก่ผู้วิจัย
- คุณลักขมิ โขมโนทัย ที่คอยช่วยเหลือในด้านต่าง ๆ และเป็นกำลังใจในยามที่ผู้วิจัยเกิดปัญหา

สุดท้ายนี้ ผู้วิจัยขอขอบคุณคณาจารย์ทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ต่าง ๆ ทั้งในอดีต และปัจจุบัน และขอกราบขอบพระคุณบิดา มารดา ที่ให้ความรัก กำลังใจ การอบรมเลี้ยงดูและส่งเสริมการศึกษาเป็นอย่างดีมาโดยตลอด รวมถึงเป็นกำลังใจอันยิ่งใหญ่แก่ผู้วิจัย จนทำให้ผู้วิจัยประสบความสำเร็จในชีวิตเรื่อยมา

นริศ มิ่งโมรา

สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย)	ก
บทคัดย่อ (ภาษาอังกฤษ)	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ช
สารบัญรูป	ญ
บทที่	
1 บทนำ	1
1.1 ความสำคัญและที่มาของปัญหาการวิจัย	1
1.2 วัตถุประสงค์ของการวิจัย	2
1.3 ขอบเขตของงานวิจัย	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
2 ปรัชญ์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง	4
2.1 วิศวกรรมความต้องการคืออะไร	4
2.1.1 กระบวนการของวิศวกรรมความต้องการ	4
2.1.2 การสกัดความต้องการ	5
2.1.3 การวิเคราะห์และต่อรองความต้องการ	7
2.1.4 เอกสารความต้องการ	9
2.1.5 การทวนสอบความต้องการ	10
2.1.6 การจัดการความต้องการ	12
2.2 แนวความคิดเกี่ยวกับการทดสอบ	13
2.2.1 ระดับของการทดสอบ	13
2.2.2 กรณีสอบ	16
2.2.3 มาตรฐาน IEEE Std 829-1998	17

สารบัญ (ต่อ)

หน้า

2.3 งานวิจัยที่เกี่ยวข้อง	21
2.3.1 งานวิจัยเกี่ยวกับการประเมินผลมาตรฐานเอกสารข้อกำหนด ความต้องการซอฟต์แวร์ทั้ง 4 มาตรฐาน	21
2.3.2 ระบบจัดการกรณีทดสอบ (Test case management system)	25
2.3.3 TestDirector 6	25
2.3.4 พัลส์วันเทส (+1 TEST)	26
3 ระเบียบวิธีวิจัย	27
3.1 ระเบียบวิธีวิจัย	27
3.2 การวิเคราะห์และออกแบบเครื่องมือช่วยในการสร้างกรณีทดสอบ สำหรับการทดสอบการยอมรับ	28
3.2.1 ส่วนรับข้อกำหนดความต้องการซอฟต์แวร์	28
3.2.2 ส่วนรับข้อมูลกรณีทดสอบ	29
3.2.3 ฐานข้อมูลกรณีทดสอบ	30
3.2.4 การวิเคราะห์และออกแบบเครื่องมือ	31
3.3 การออกแบบและสร้างฐานข้อมูลสำหรับเครื่องมือช่วยในการสร้าง กรณีทดสอบสำหรับการทดสอบการยอมรับ	32
3.4 การออกแบบส่วนนำเข้าข้อมูล	75
4 การทดสอบและอภิปรายผล	91
4.1 สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือสร้างกรณีทดสอบ	91
4.2 โครงสร้างของเครื่องมือ	92
4.3 กรณีทดสอบที่ใช้ทดสอบเครื่องมือ	92
4.4 ขั้นตอนการทดสอบ	93
4.5 ผลการทดสอบโปรแกรม	93
4.6 ขั้นตอนการสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ	94
4.6 อภิปรายผล	105

สารบัญ (ต่อ)

หน้า

5 บทสรุป	106
5.1 สรุปผลการวิจัย	106
5.2 ประโยชน์ของเครื่องมือสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ	107
5.3 ข้อจำกัดของเครื่องมือที่สร้างขึ้น	107
5.4 แนวทางในการพัฒนาต่อ	107
รายการอ้างอิง	109
ภาคผนวก	
ภาคผนวก ก บทความผลงานวิจัยที่นำเสนอในการประชุมวิชาการวิทยาศาสตร์ และเทคโนโลยีแห่งประเทศไทย ครั้งที่ 32	110
ภาคผนวก ข ตัวอย่างกรณีทดสอบ ของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติ	129
ภาคผนวก ค คู่มือการใช้งานเครื่องมือสร้างกรณีทดสอบสำหรับการ ทดสอบการยอมรับ	143
ประวัติผู้เขียน	162

สารบัญตาราง

ตารางที่	หน้า
2.1 โครงสร้างของเอกสารความต้องการตามมาตรฐาน IEEE	9
2.2 การประเมินผลของมาตรฐาน	24
3.1 รายละเอียดในตาราง Emp	32
3.2 รายละเอียดในตาราง Customer	33
3.3 รายละเอียดในตาราง Cust_release	33
3.4 รายละเอียดในตาราง Project	34
3.5 รายละเอียดในตาราง Project_head.....	34
3.6 รายละเอียดในตาราง Sec01_Abbreviations	35
3.7 รายละเอียดในตาราง Sec01_Acronyms	36
3.8 รายละเอียดในตาราง Sec01_Definition.....	36
3.9 รายละเอียดในตาราง Sec01_Overview	37
3.10 รายละเอียดในตาราง Sec01_Purpose	38
3.11 รายละเอียดในตาราง Sec01_References	38
3.12 รายละเอียดในตาราง Sec01_Scope	39
3.13 รายละเอียดในตาราง Sec02_Apport_Req	40
3.14 รายละเอียดในตาราง Sec02_Assump_Depend	41
3.15 รายละเอียดในตาราง Sec02_Comm_Interface.....	42
3.16 รายละเอียดในตาราง Sec02_Constraint	43
3.17 รายละเอียดในตาราง Sec02_Hard_Interface.....	44
3.18 รายละเอียดในตาราง Sec02_Mem_Constraint.....	44
3.19 รายละเอียดในตาราง Sec02_Operation	45
3.20 รายละเอียดในตาราง Sec02_Prod_Func	46
3.21 รายละเอียดในตาราง Sec02_Qua_Prov.....	47
3.22 รายละเอียดในตาราง Sec02_Req_Trace	48
3.23 รายละเอียดในตาราง Sec02_Site_Adap_Req.....	49

สารบัญตาราง (ต่อ)

ตารางที่	หน้า
3.24 รายละเอียดในตาราง Sec02_Soft_Interface	50
3.25 รายละเอียดในตาราง Sec02_User_Interface	50
3.26 รายละเอียดในตาราง Sec03_Acceptance_Req.....	51
3.27 รายละเอียดในตาราง Sec03_Availability.....	52
3.28 รายละเอียดในตาราง Sec03_Comp_Hard_Resource_Req.....	53
3.29 รายละเอียดในตาราง Sec03_Doc_Req.....	54
3.30 รายละเอียดในตาราง Sec03_Ext_Interface	55
3.31 รายละเอียดในตาราง Sec03_Functional_Requirement	56
3.32 รายละเอียดในตาราง Sec03_Implementation_Constrain	57
3.33 รายละเอียดในตาราง Sec03_Logic_Dat_Req	58
3.34 รายละเอียดในตาราง Sec03_Logistic_Related_Req	59
3.35 รายละเอียดในตาราง Sec03_Maintainability	60
3.36 รายละเอียดในตาราง Sec03_Operate_Req.....	61
3.37 รายละเอียดในตาราง Sec03_Packaging_Req.....	62
3.38 รายละเอียดในตาราง Sec03_Perform_Req	63
3.39 รายละเอียดในตาราง Sec03_Portability	64
3.40 รายละเอียดในตาราง Sec03_Reliability	65
3.41 รายละเอียดในตาราง Sec03_Safety_Req	66
3.42 รายละเอียดในตาราง Sec03_Security.....	67
3.43 รายละเอียดในตาราง Sec03_Software_Quality_Factor	68
3.44 รายละเอียดในตาราง Sec03_Standard_Compliance	69
3.45 รายละเอียดในตาราง Sec03_Testcase	69
3.46 รายละเอียดในตาราง Sec03_Testcase_Input.....	70
3.47 รายละเอียดในตาราง Sec03_Training_Releated_Req.....	71
3.48 รายละเอียดในตาราง Sec03_Verification_Req	72

สารบัญตาราง (ต่อ)

ตารางที่	หน้า
3.49 รายละเอียดในตาราง Sec04_Appendix	73
3.50 รายละเอียดในตาราง Sec04_Index.....	73
3.51 รายละเอียดในตาราง Sec04_Note	74
3.52 รายละเอียดในตาราง Sec04_Table_Of_Content.....	75

สารบัญรูป

รูปที่	หน้า
2.1 โมเดลของ Requirements engineering process	5
2.2 กลไกการสกัดความต้องการ (Generic requirements elicitation process).....	6
2.3 Requirements analysis and negotiation process.....	8
2.4 Requirements validation process	11
2.5 Requirement management.....	12
2.6 V-Model.....	16
2.7 ความสัมพันธ์ของเอกสารที่ใช้ในการทดสอบตามมาตรฐาน IEEE	18
3.1 องค์ประกอบหลักของเครื่องมือ	28
3.2 แผนผังการทำงานของส่วนรับข้อกำหนดความต้องการซอฟต์แวร์	29
3.3 แสดงลำดับขั้นตอนการสร้างกรณีทดสอบ	30
3.4 แผนภาพกระแสข้อมูลของเครื่องมือ	31
3.5 หน้าจอเมนูหลัก.....	77
3.6 หน้าจอบันทึกรายละเอียดพนักงาน	78
3.7 หน้าจอบันทึกรายละเอียดลูกค้า.....	79
3.8 หน้าจอบันทึกรายละเอียด Release	80
3.9 หน้าจอบันทึกรายละเอียดโครงการ	81
3.10 หน้าจอบันทึกข้อกำหนดความต้องการซอฟต์แวร์.....	83
3.11 หน้าจอบันทึกข้อกำหนดความต้องการซอฟต์แวร์.....	84
3.12 หน้าจอบันทึกข้อกำหนดความต้องการซอฟต์แวร์.....	85
3.13 หน้าจอบันทึกข้อกำหนดความต้องการซอฟต์แวร์.....	86
3.14 หน้าจอบันทึกข้อกำหนดความต้องการซอฟต์แวร์.....	87
3.15 หน้าจอบันทึกข้อกำหนดความต้องการซอฟต์แวร์.....	88
3.16 หน้าจอบันทึกข้อกำหนดความต้องการซอฟต์แวร์.....	89
3.17 หน้าจอบันทึกกรณีทดสอบ	90
4.1 ตัวต้นแบบในส่วน of ข้อกำหนดความต้องการซอฟต์แวร์.....	95

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.2	บันทึกรายละเอียดพนักงาน98
4.3	บันทึกรายละเอียดลูกค้า.....99
4.4	บันทึกรายละเอียด Release100
4.5	บันทึกรายละเอียดโครงการ101
4.6	บันทึกข้อกำหนดความต้องการซอฟต์แวร์102
4.7	บันทึกในส่วนของการทดสอบ103
4.8	พิมพ์เอกสารการทดสอบ.....104
4.9	เอกสารการทดสอบ105

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของปัญหาการวิจัย

ในการพัฒนาซอฟต์แวร์ ขั้นตอนหนึ่งที่สำคัญเพื่อให้ซอฟต์แวร์ที่พัฒนาขึ้นนั้น มีประสิทธิภาพน่าเชื่อถือ และถูกต้องตามความต้องการของผู้ใช้งานคือ ขั้นตอนการทดสอบ ซึ่งขั้นตอนสุดท้ายของการทดสอบก่อนการส่งมอบงานอย่างเป็นทางการให้กับผู้ใช้งานคือ การทดสอบการยอมรับ โดยจะมีวิธีการทดสอบแบบแบล็กบ็อกซ์ เพื่อให้เป็นไปตามความต้องการทางธุรกิจที่ได้ระบุไว้ในข้อกำหนดความต้องการซอฟต์แวร์ โดยที่ผู้ใช้หรือกระบวนการต่าง ๆ นั้นจะทำตามระบบงานที่ได้ออกแบบไว้ สำหรับการทดสอบการยอมรับ สิ่งที่เป็นสำหรับผู้ทดสอบเพื่อใช้ในการทดสอบคือ กรณีทดสอบ ซึ่งการออกแบบกรณีทดสอบเป็นเรื่องที่ยุ่งยากและใช้เวลามากสำหรับผู้ทดสอบ

สำหรับความต้องการเชิงหน้าที่ (Functional Requirements) หมายถึง การทำเอกสารใน Software Requirement Specification (SRS) ซึ่งจะบรรยายถึงความเป็นไปได้ของความคาดหวังเกี่ยวกับพฤติกรรมภายนอกของระบบซอฟต์แวร์ ซึ่งจะถูกใช้ในการพัฒนา, การตรวจสอบ, การรับประกันคุณภาพ, การจัดการโครงการ และความสัมพันธ์ของฟังก์ชันในโครงการ

ส่วนการทดสอบการยอมรับวัตถุประสงค์ของการทดสอบการยอมรับ เพื่อยืนยันว่า Application ที่ทดสอบนั้นสามารถทำงานได้ตรงตามความต้องการ (Requirement) และสร้างความเชื่อมั่นว่าระบบสามารถทำงานได้ถูกต้องก่อนที่จะส่งมอบอย่างเป็นทางการให้แก่ End User

ซึ่งการทดสอบการยอมรับ (Acceptance Testing) จะสัมพันธ์กับช่วงของ Requirements Phase ของ V Model ซึ่ง Phase ของการทำ Requirements นั้นจะต้องมีการวางแผน การทบทวนเพื่อตรวจสอบ Requirement และกำหนดตัวแทนของผู้ใช้งาน ไว้ล่วงหน้าก่อนทำการทดสอบการยอมรับ

ดังนั้นงานวิจัยนี้จึงมีแนวคิดในการออกแบบและพัฒนาเครื่องมือช่วยในการสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ เพื่อนำมาช่วยผู้ทดสอบในการสร้างกรณีทดสอบให้ครอบคลุมเนื้อหาที่ระบุไว้ในข้อกำหนดความต้องการซอฟต์แวร์ และนอกจากนั้น ยังช่วยในการจัดเก็บข้อมูลกรณีทดสอบที่ใช้ในการทดสอบลงฐานข้อมูล เพื่อประโยชน์ในการนำกลับมาใช้ใหม่ และเป็นการลดภาระของผู้ทดสอบอีกด้วย โดยในงานวิจัยนี้จะศึกษาถึงวิธีการในการจัดเก็บ

ความต้องการเชิงหน้าที่ ที่ได้มาจากข้อกำหนดความต้องการซอฟต์แวร์ โดยอ้างอิงมาจากมาตรฐานต่างประเทศ 4 มาตรฐาน ได้แก่ มาตรฐาน IEEE Std.830-1993, มาตรฐาน ESA PSS-05-0, มาตรฐาน NASA-DID-P200 และมาตรฐาน DOD-STD-498 และนำเอาความต้องการเชิงหน้าที่ ที่ได้มาจากข้อกำหนดความต้องการซอฟต์แวร์ มาสร้างเป็นกรณีทดสอบในส่วนของทดสอบการยอมรับ โดยกรณีทดสอบที่ได้จะครอบคลุมและตรงตามความต้องการของผู้ใช้งานมากขึ้น ส่วนการจัดเก็บข้อมูลกรณีทดสอบ และผลการทดสอบภายในระบบ จะอ้างอิงตามมาตรฐานของ IEEE Std.829-1998

1.2 วัตถุประสงค์ของการวิจัย

เพื่อพัฒนาเครื่องมือที่สามารถนำเอาข้อมูลในส่วนของความต้องการเชิงหน้าที่ มาสร้างกรณีทดสอบ (Test Case) สำหรับการทดสอบการยอมรับ และจัดทำฐานข้อมูลสำหรับเก็บข้อมูลกรณีทดสอบได้

1.3 ขอบเขตของงานวิจัย

- 1.3.1 สร้างเอกสารข้อกำหนดความต้องการซอฟต์แวร์ที่อ้างอิงกับมาตรฐาน 4 มาตรฐาน ได้แก่ IEEE Std 830-1998, ESA PSS-05-0, NASA-DID-P200 และ DoD-STD-498
- 1.3.2 การสร้างกรณีทดสอบ จะสร้างมาจากความต้องการเชิงหน้าที่เท่านั้น
- 1.3.3 การใส่ข้อมูลความต้องการที่ใช้ในระบบได้มาจากเอกสารข้อกำหนดความต้องการ
- 1.3.4 กรณีทดสอบที่พัฒนาขึ้น และใช้งานภายในระบบ มีรูปแบบที่ปรับปรุงจากมาตรฐานของ IEEE Std 829-1998
- 1.3.5 ในส่วนการสร้างกรณีทดสอบ ระบบจะรับประเภทข้อมูล และลักษณะเฉพาะ หรือเงื่อนไขของข้อมูลที่ได้จากเอกสารข้อกำหนดความต้องการซอฟต์แวร์ที่อ้างอิงกับมาตรฐาน 4 มาตรฐาน มาใช้ในการสร้างกรณีทดสอบ

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1.4.1 ทำให้องค์กรที่ทำการพัฒนาซอฟต์แวร์สามารถนำเครื่องมือนี้ไปใช้ในการเก็บรวบรวมข้อกำหนดความต้องการที่เกิดขึ้นเป็นมาตรฐานเดียวกันทั้งองค์กร
- 1.4.2 ช่วยลดภาระผู้ทดสอบ เนื่องจากระบบสามารถสร้างกรณีทดสอบที่ครอบคลุมเนื้อหาที่ระบุอยู่ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ สำหรับวิธีการแบบสื่อกบอกรู้ได้โดยอัตโนมัติ

- 1.4.3 มีการเก็บข้อมูลกรณีทดสอบที่เป็นมาตรฐานเดียวกันทำให้ผู้ทดสอบสามารถใช้งานหรือแก้ไขปรับปรุงกรณีทดสอบได้ง่ายขึ้น
- 1.4.4 กรณีทดสอบที่บันทึกไว้สามารถนำไปใช้ในการทดสอบซอฟต์แวร์ที่จะพัฒนาขึ้นมาใหม่ในอนาคตได้ สามารถนำเครื่องมือที่พัฒนาขึ้นไปใช้งานร่วมกับเครื่องมือที่ทำงานเกี่ยวข้องกันได้ เช่น เครื่องมือที่ช่วยวางแผนการทดสอบ

บทที่ 2

ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

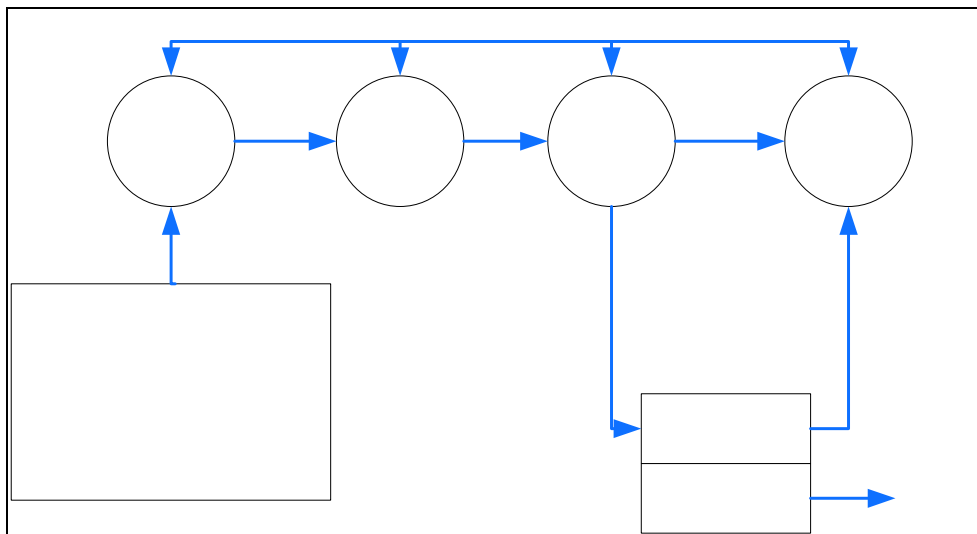
2.1 วิศวกรรมความต้องการคืออะไร

การรวบรวมข้อมูลความต้องการต่าง ๆ ระบบ ไม่ว่าจะเป็น ข้อกำหนดหรือความต้องการต่าง ๆ ของผู้ใช้งาน เอกสารข้อมูล ฮาร์ดแวร์ ข้อบังคับขององค์กร รวมทั้งกฎหมาย ในรูปแบบของ “Engineering” ซึ่งคำว่า “Engineering” ในที่นี้หมายถึง เป็นระบบ และมีการกระทำแบบซ้ำ

วิศวกรรมความต้องการ มีความคล้ายคลึงหลายอย่างกับ การวิเคราะห์ระบบ ในหลักการแล้ว การวิเคราะห์ระบบ จะใช้สำหรับวิเคราะห์ ระบบธุรกิจซึ่งจะเจาะจงไปในทางระบบธุรกิจมากกว่าแต่ วิศวกรรมความต้องการ จะเจาะจงทั้งในระบบธุรกิจและระบบคอมพิวเตอร์ด้วย

2.1.1 กระบวนการของวิศวกรรมความต้องการ (Requirements engineering process)

(Ian Sommerville, 1992) คือ กลุ่มของกิจกรรมที่ใช้สร้างบทกำหนดของความ ต้องการ (Requirement definition) และการระบุความต้องการ (Requirement specification) ที่บรรจุ เอกสารความต้องการ (Requirement document หรือ Software requirement document) เอกสารชุดนี้ นับเป็นจุดเริ่มต้นในการพัฒนาในขั้นตอนต่อไป (การออกแบบ, การเขียนโปรแกรม, การทดสอบ และการบำรุงรักษาระบบ) เพราะฉะนั้นข้อกำหนดความต้องการจึงต้องมีการชัดเจน สอดคล้องกัน ขั้นตอนต่าง ๆ ของกระบวนการ requirement engineering ยังไม่มีชัดเจนเนื่องจากขึ้นอยู่กับว่าแต่ละองค์กรจะมีระบบอย่างไร



รูปที่ 2.1 โมเดลของ Requirements engineering process

2.1.2 การสกัดความต้องการ (Requirements elicitation)

ในขั้นตอนนี้เราจะสามารถทราบความต้องการต่าง ๆ โดยผู้พัฒนาระบบต้องไปปรึกษากับลูกค้า จากเอกสารของระบบ ความรู้ต่าง ๆ ในหัวข้อที่เราจะศึกษา การให้บริการของระบบที่เราจะพัฒนา ความต้องการของประสิทธิภาพของระบบ ข้อจำกัดทางด้านฮาร์ดแวร์ เพื่อที่ทราบถึงแนวทางในการแก้ปัญหา

การสกัดความต้องการ เป็นส่วนที่สำคัญมาก หากเราไม่สามารถทราบถึงความต้องการที่แท้จริงของลูกค้าแล้ว เมื่อเราพัฒนาระบบออกมา อาจทำให้ลูกค้าไม่พอใจในตัวระบบองค์ประกอบของการสกัดความต้องการ ได้แก่

1) ความเข้าใจในขอบเขตของแอปพลิเคชัน (Application domain understanding) เป็นการทำความเข้าใจเกี่ยวกับขอบเขตหัวข้อของระบบที่เราจะพัฒนา ตัวอย่างเช่นถ้าเราจะพัฒนาระบบ จะทำอะไร มีขอบเขตอะไรบ้าง

2) ความเข้าใจในปัญหา (Problem understanding) เราต้องทำความเข้าใจในรายละเอียดของปัญหาของลูกค้าที่เราต้องใช้ระบบของเราจัดการนั้น ต้องเข้าใจว่าปัญหาหรืออุปสรรคใด ๆ ในระบบธุรกิจหรืองานที่ต้องการนำเอาระบบคอมพิวเตอร์ไปแก้ไข คืออะไร

3) เข้าใจในธุรกิจหรือระบบที่ศึกษาอยู่ (Business understanding) เราต้องเข้าใจว่าระบบจะทำการปฏิสัมพันธ์ และมีผลต่อระบบธุรกิจในแต่ละส่วนอย่างไร เราต้องเข้าใจว่าช่วยเหลือเป้าหมายในทางธุรกิจของบริษัททั้งหมดได้อย่างไร

Requirements

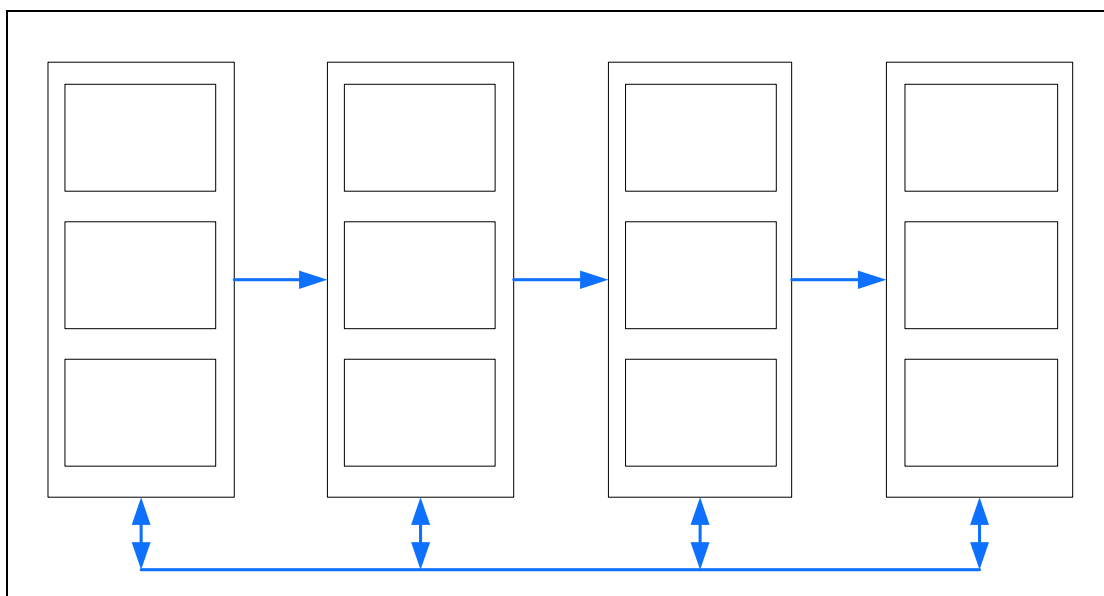
elicitation

User needs domain

existing system in

regulations, stan

4) ความเข้าใจความต้องการ และเงื่อนไขของผู้ที่มีส่วนได้ส่วนเสียในระบบ (Understanding the needs and constraints of system stakeholders) ซึ่งในระบบจะมีผู้เกี่ยวข้องหลายฝ่าย แต่ละฝ่ายต้องการอะไร มีเงื่อนไขอะไร



รูปที่ 2.2 กลไกการสกัดความต้องการ (Generic requirements elicitation process)

Requirements elicitation process ประกอบไปด้วยกิจกรรมที่สำคัญทั้งหมด 4 อย่างได้แก่ (Gerald and Sommerville, 2002)

1) การตั้งจุดมุ่งหมาย (Objective setting) วัตถุประสงค์ขององค์กร ซึ่งประกอบไปด้วย เป้าหมายหลักของธุรกิจ และการบรรยายถึงปัญหาที่ต้องการนำมาแก้ไข และข้อจำกัดของระบบเช่น งบประมาณ เวลา ควรจะถูกกำหนดออกมาในขั้นตอนนี้

2) การรู้ข้อมูลเบื้องหลังของระบบ (Background knowledge acquisition) เป็นขั้นตอนที่สำคัญมาก ๆ ซึ่งเป็นขั้นตอนที่วิศวกรความต้องการจะรวบรวมและเข้าใจในข้อมูลพื้นฐานต่าง ๆ ของระบบ และส่วนนี้ยังใช้บอกได้ว่า ส่วนใดขององค์กรที่ระบบจะถูกนำไปใช้ ข้อมูล application domain ของระบบ และระบบที่มีอยู่แล้วในองค์กรซึ่งกำลังใช้อยู่และระบบที่จะเอาระบบที่เราพัฒนาเข้าไปแทนที่

3) การมีความรู้เกี่ยวกับภาพรวมขององค์กร (Knowledge organization) ข้อมูลและองค์ความรู้ทั้งหมดที่ถูกเก็บมาใน 2 ส่วนที่แล้วจะถูกทำให้เป็นระบบและจัดเรียง ซึ่งได้แก่ ผู้ที่มี

Establish objectives

Business goals

ส่วนเกี่ยวข้องในระบบ และบทบาทของบุคคลเหล่านั้น เรียงลำดับความสำคัญเป้าหมายขององค์กร และ tacit domain knowledge ที่ไม่เกี่ยวข้องโดยตรงกับความต้องการของระบบ

4) การรวบรวมความต้องการของผู้มีส่วนได้ส่วนเสียกับระบบ (Stakeholder requirements collection) เป็นส่วนที่เรียกว่าการดึงข้อมูลออกมาจริง ๆ ซึ่งรวมไปทั้งการปรึกษาผู้ที่เกี่ยวข้องในระบบเกี่ยวกับสิ่งที่บุคคลเหล่านั้นต้องการและการได้มาซึ่งความต้องการต่าง ๆ ที่มาจาก application domain และจากองค์กรที่ต้องการระบบของเราไปใช้งาน

ผลลัพธ์ของ Requirements elicitation process ควรจะออกมาในรูปแบบของเอกสารความต้องการต้นแบบ (Draft requirements document) ที่บรรยายถึงความต้องการของระบบ ซึ่งต่อไปเอกสารนี้จะถูกนำไปวิเคราะห์ เพื่อที่จะค้นหาปัญหาและข้อขัดแย้งของความต้องการต่าง ๆ ของผู้ใช้ ปัญหาการขัดแย้งของความต้องการ กับปัญหาการทับซ้อนกันของความต้องการเป็นปัญหาที่ไม่สามารถหลบเลี่ยงได้ ดังนั้นจึงต้องมีการต่อรอง เพื่อที่จะแก้ปัญหาดังกล่าวเพิ่มเติมอีกด้วย

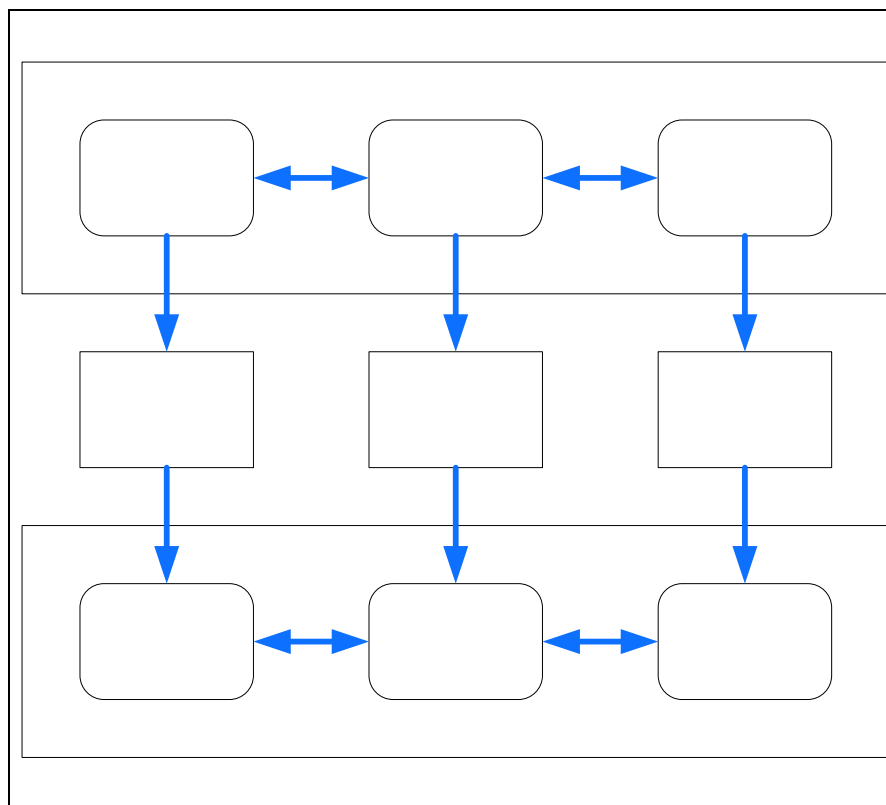
2.1.3 การวิเคราะห์และต่อรองความต้องการ (Requirements analysis and negotiation)

เป้าหมายของการวิเคราะห์ความต้องการของผู้ใช้ก็คือ เพื่อที่จะค้นหาปัญหาในเอกสารความต้องการต้นแบบ ขณะดำเนินการวิเคราะห์ความต้องการต่าง ๆ นักวิเคราะห์ต้องพยายามทำความเข้าใจกับปัญหาต่าง ๆ ขององค์กร

ขั้นตอนต่าง ๆ ของ Requirements analysis process จะมีกิจกรรมที่ต้องดำเนินการเข้าไปซ้ำมาจนกระทั่งได้ข้อมูลที่ต้องการและชัดเจนที่สุด ซึ่งขั้นตอนนี้มีกิจกรรมต่าง ๆ ดังนี้

- การตรวจสอบความต้องการ (Necessity checking) สิ่งที่เป็นสำหรับความต้องการต่าง ๆ จะถูกวิเคราะห์ ซึ่งในบางกรณี ความต้องการบางอย่างอาจจะไม่ได้เกี่ยวข้องกับเป้าหมายทางธุรกิจขององค์กร
- การตรวจสอบความสอดคล้องและสมบูรณ์ของความต้องการ (Consistency and completeness checking) ความต้องการจะถูกตรวจสอบ (cross-checked) สำหรับความสอดคล้องและความสมบูรณ์ ความสอดคล้องหมายความว่าต้องไม่มีความต้องการใดที่ขัดแย้งกัน ส่วนความสมบูรณ์หมายความว่า จะต้องไม่มีบริการหรือข้อกำหนดใด ๆ ตกหล่นหรือขาดหายไป
- การตรวจสอบความเป็นไปได้ (Feasibility checking) ความต้องการจะถูกตรวจสอบเพื่อให้แน่ใจว่าโครงการสามารถจะดำเนินการไปได้ ภายใต้งบประมาณ และเวลาที่กำหนดไว้

กิจกรรมเหล่านี้จะให้ผลลัพธ์ออกมาในรูปแบบของการจำแนกเซตของความต้องการ ซึ่งจะถูกนำเอาไปใช้ในกระบวนการต่อรอง (Requirements negotiation process) โดยผลลัพธ์เหล่านี้จะบรรจุกาต่าง ๆ ที่ได้จากขั้นตอนการวิเคราะห์ความต้องการ



Requirements

รูปที่ 2.3 requirements analysis and negotiation process

ขั้นตอนวิธีของกระบวนการต่อรอง มีดังนี้คือ

- การตัดสินใจในความต้องการ (Requirements discussion) ความต้องการที่เป็นปัญหาจะต้องนำไปปรึกษากับผู้เกี่ยวข้องในระบบ ให้พวกเขาได้บ่งบอกถึงความต้องการนั้น ๆ ในแง่มุมของเขาให้เราเข้าใจ
- ลำดับความสำคัญของความต้องการ (Requirements prioritization) ความต้องการที่ขัดแย้งกัน จะถูกจัดลำดับความสำคัญ เพื่อที่จะกำหนดความต้องการวิกฤต เพื่อช่วยในกระบวนการตัดสินใจ
- ข้อตกลงของความต้องการ (Requirements agreement) วิธีแก้ปัญหสำหรับความต้องการต่าง ๆ จะถูกกำหนดและ เซตของการตกลงกันได้ต่าง ๆ จะถูกยอมรับโดยทั่วไปแล้ว นี่จะเป็นส่วนหนึ่งที่เกี่ยวข้องกับการเปลี่ยนความต้องการต่าง ๆ สำหรับบางความต้องการ

Ne
ch

Unn
requ

2.1.4 เอกสารความต้องการ (Requirements document)

เอกสาร Requirements document ไม่ได้เป็นเอกสารที่ใช้ในการออกแบบระบบซอฟต์แวร์ ดังนั้นจึงไม่มีการระบุรายละเอียดวิธีการออกแบบว่าต้องทำอะไร หากแต่เป็นสิ่งที่ระบุว่าควรมีอะไรบ้างในซอฟต์แวร์ที่กำลังพัฒนา เอกสารชุดนี้เป็นเครื่องมือสำหรับติดตามการดำเนินงานระหว่างช่วงการค้นหาคำความต้องการและการออกแบบในขั้นสุดท้าย ซึ่งหมายความว่าความต้องการต่าง ๆ จะถูกนำมาแปลงเป็นข้อมูลที่ใช้ในการออกแบบระบบและเขียนโปรแกรม

หลักการเขียนรายละเอียดที่อยู่ในเอกสารชุดนี้คือ จะต้องมีความสมบูรณ์และข้อมูลต่าง ๆ ในเอกสารต้องไม่ขัดแย้งกัน คุณสมบัติที่ควรมีใน Requirements document คือ

- ควรมีพฤติกรรมภายนอกที่เกี่ยวข้องกับระบบ
- ควรมีการระบุเงื่อนไขหรือข้อจำกัดภายในระบบ
- ควรอยู่ในรูปแบบที่ง่ายสำหรับการแก้ไข
- สามารถนำมาเป็นเครื่องมืออ้างอิงในระบบ
- ควรมีการบันทึกรายละเอียดขั้นตอนการทำงานที่เกิดขึ้น รวมถึงวัฏจักรการพัฒนา
- ควรมีการระบุวิธีการทำงานของระบบ เมื่อมีเหตุการณ์ที่ไม่ต้องการเกิดขึ้น

เอกสารความต้องการประกอบไปด้วย 2 ส่วนคือ Requirements definition และ Requirements specification โครงสร้างที่ดีควรจัดแบ่งรายละเอียดออกเป็นบท ๆ และมีการอธิบายศัพท์ต่าง ๆ ที่ใช้ในเอกสาร

ตารางที่ 2.1 โครงสร้างของ Requirement document ตามมาตรฐาน IEEE

บทที่	รายละเอียด
Introduction	อธิบายถึงสิ่งต่าง ๆ ที่จะมีในระบบ และมีการระบุประโยชน์ที่ได้รับ หน้าที่ต่าง ๆ ที่มีในระบบ เป้าหมายของการพัฒนา วิธีการหรือยุทธวิธีที่จะนำไปใช้ในองค์กร
Glossary	อธิบายศัพท์ทางเทคนิคที่มีในเอกสาร ซึ่งในการเขียนคำอธิบายจะตั้งสมมติฐานว่าผู้อ่านประกอบด้วยผู้มีและไม่มีประสบการณ์

ตารางที่ 2.2 โครงสร้างของ Requirement document ตามมาตรฐาน IEEE (ต่อ)

บทที่	รายละเอียด
System model	ตัวแบบระบบซึ่งแสดงความสัมพันธ์ของระบบกับสภาพแวดล้อมภายนอก ว่าติดต่อกับองค์กรหรือระบบใดบ้าง มีกระบวนการทำงานอย่างไร
Functional requirement definition	อธิบายบริการต่าง ๆ ที่ผู้ใช้ต้องการมีในระบบใหม่ ซึ่งอาจใช้ภาษาธรรมชาติ แผนภูมิหรือสัญลักษณ์ที่ผู้ใช้สามารถอ่านเข้าใจได้ง่าย
Non-Functional requirement definition	รายละเอียดข้อจำกัดและเงื่อนไขต่าง ๆ ที่ระบบต้องพัฒนา เช่น เวลา ขนาด หน่วยความจำ
System evolution	รายละเอียดของสมมติฐานที่คาดว่าจะอาจมีการเปลี่ยนแปลงในระบบ เช่น ความต้องการ วิวัฒนาการของฮาร์ดแวร์
Requirement specification	ระบุรายละเอียด Functional requirement definition specification

สำหรับความต้องการซอฟต์แวร์ (Software Requirements) นั้นหมายถึงคุณลักษณะหรือความสามารถของซอฟต์แวร์ที่ถูกกำหนด เพื่อใช้สำหรับแก้ปัญหาหรือดำเนินงานให้บรรลุเป้าหมายตามความต้องการของผู้ใช้ โดยที่ความสามารถเหล่านั้นจะถูกกำหนดในรูปของสัญญาข้อกำหนด มาตรฐาน หรือเอกสารมาตรฐานอื่นที่กำหนด โดยที่ความต้องการนั้นสามารถแบ่งออกเป็น 2 ประการคือ

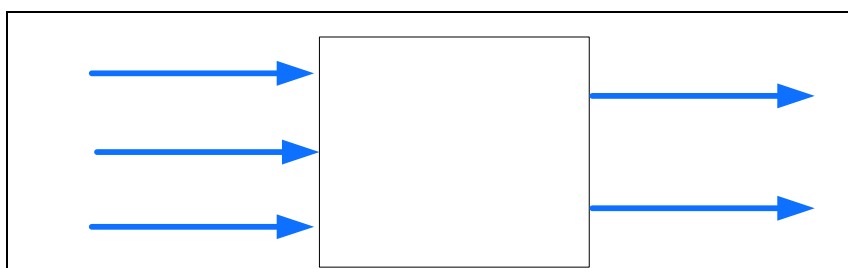
- ความต้องการเชิงหน้าที่ (Functional Requirements) คือ การทำเอกสารในข้อกำหนดความต้องการซอฟต์แวร์ (Software Requirements Specification - SRS) ซึ่งจะบรรยายถึงความเป็นไปได้ของความคาดหวังเกี่ยวกับพฤติกรรมภายนอกของระบบซอฟต์แวร์ ซึ่งจะถูกใช้ในการพัฒนา ตรวจสอบ ประกันคุณภาพ การจัดการโครงการ และความสัมพันธ์ของฟังก์ชันในโครงการ
- ความต้องการที่ไม่ใช่ความต้องการเชิงหน้าที่ (Non-Functional Requirements) คือ สิ่งที่กำหนดถึงประสิทธิภาพของซอฟต์แวร์ ข้อบังคับ และมาตรฐานที่ใช้ในระหว่างการพัฒนาซอฟต์แวร์

2.1.5 การทวนสอบความต้องการ (Requirements validation)

เป็นขั้นตอนสุดท้ายในกระบวนการ Requirements engineering ซึ่งมีเป้าหมายคือเพื่อให้ความต้องการที่เราเก็บมาถูกต้อง โดย Requirements validation จะมุ่งเน้นไปในทางการตรวจสอบ เอกสารความต้องการ (Requirements document) ซึ่งรวบรวมความต้องการทุกอย่างของ

ระบบและ ความไม่สมบูรณ์พร้อมทั้งความซ้อนทับของความต้องการถูกกำจัดไปทั้งหมดแล้ว นอกจากนี้การทำ requirements validation เราควรมีคำถามคำถามหนึ่งอยู่ในใจเสมอ นั่นคือ เราได้ความต้องการที่ถูกต้องแล้วหรือยัง เกณฑ์การตรวจสอบความต้องการมี 4 ขั้นตอนคือ

- ความถูกต้อง (Validity) ความต้องการต่าง ๆ ที่ถูกคำระบุในเอกสารต้องตรงกับสิ่งที่ลูกค้าต้องการจริง ๆ
- ความสอดคล้อง (Consistency) ความต้องการต่าง ๆ ต้องไม่ขัดแย้งซึ่งกันและกัน
- ความสมบูรณ์ (Completeness) รายละเอียดต่าง ๆ ที่เป็นความต้องการและเงื่อนไขกฎเกณฑ์ทั้งหมด จะต้องถูกระบุอยู่ในเอกสาร
- ความเป็นจริง (Realism) ในทางปฏิบัติไม่มีทางใดที่จะบอกได้ว่า ความต้องการต่าง ๆ ที่ระบุในข้อกำหนดสามารถนำมาพัฒนาระบบได้จริง



รูปที่ 2.4 Requirements validation process

Input ของระบบ Requirements validation process มีดังนี้

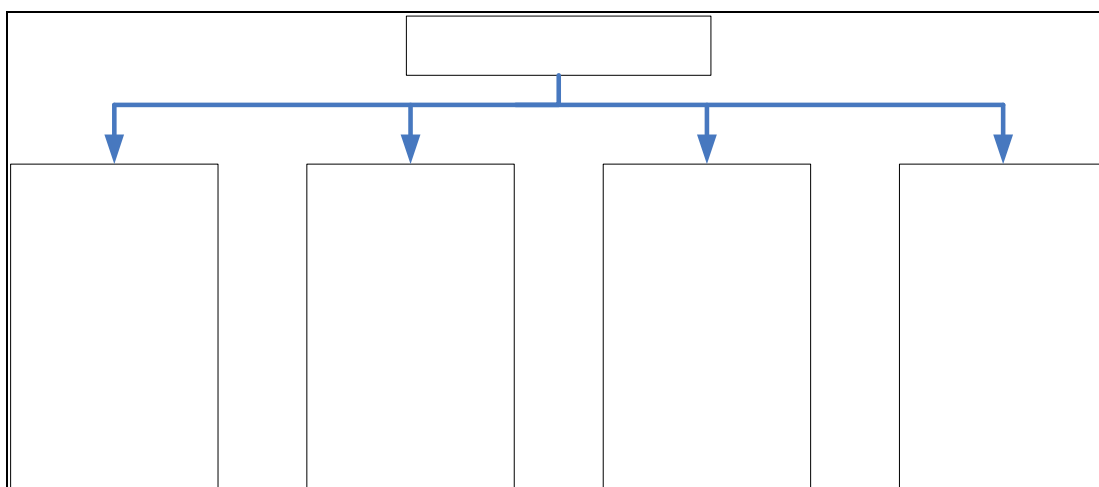
- เอกสารความต้องการ (Requirements document) ส่วนนี้ควรจะเป็นเอกสารที่สมบูรณ์แล้วซึ่งถูกจัดการให้อยู่ในรูปแบบที่ตรงมาตรฐานขององค์กรนั้น
- มาตรฐานขององค์กร (Organizational standards) กระบวนการตรวจสอบต้องสอดคล้องกับมาตรฐานขององค์กร ดังนั้น มาตรฐานขององค์กรจึงเป็นส่วนหนึ่งของ input ของกระบวนการนี้
- ความรู้ที่เกี่ยวกับองค์กร (Organizational knowledge) นี้ไม่ได้เป็น input ที่สามารถสัมผัสได้ แต่เป็นส่วนที่สำคัญ บุคคลที่ทำกระบวนการ requirements validation ควรมีความรู้ในองค์กรที่จะใช้ระบบที่พัฒนาด้วยเพื่อประโยชน์ในการตรวจสอบความถูกต้อง

Output ที่ได้จากกระบวนการ Requirements validation ได้แก่

- รายการปัญหา (Problem list) เป็นส่วนที่รายงานปัญหาที่ได้มาจากเอกสารความต้องการ (Requirements document) ซึ่งเราควรจะต้องเรียงปัญหาต่าง ๆ ที่ได้ออกมาเป็นหมวดหมู่
- ข้อตกลงในกิจกรรม (Agreed actions) ส่วนนี้เป็นรายการของการกระทำต่าง ๆ ที่จะตอบสนองต่อปัญหาที่เราเจอในกระบวนการ ซึ่งไม่เสมอไปที่จะมีการกระทำต่าง ๆ กับแก้ปัญหาแบบ 1:1 บางปัญหาอาจจะต้องใช้หลายการกระทำเพื่อที่จะทำให้สามารถแก้ปัญหาได้

2.1.6 การจัดการความต้องการ (Requirements management)

Requirements management เป็นกระบวนการจัดการความเปลี่ยนแปลงที่มีต่อระบบความต้องการ ความต้องการมีความเปลี่ยนแปลงได้เนื่องจาก การเปลี่ยนแปลงของสิ่งแวดล้อมในระบบมีการเปลี่ยนแปลง และลูกค้าสามารถพัฒนาความเข้าใจในความต้องการของตัวเองได้มากขึ้น กิจกรรมในการจัดการความต้องการมี 4 ส่วนดังนี้



รูปที่ 2.5 Requirement management

2.2 แนวความคิดเกี่ยวกับการทดสอบ

ในกระบวนการพัฒนาซอฟต์แวร์จุดมุ่งหมายหลักคือ การที่กระบวนการพัฒนาซอฟต์แวร์สามารถผลิตผลิตภัณฑ์ซอฟต์แวร์ต่าง ๆ ได้อย่างถูกต้องและตรงตามความต้องการของลูกค้าหรือผู้ใช้งาน ดังนั้นในการพัฒนาซอฟต์แวร์จึงจำเป็นต้องมีกระบวนการสำคัญต่าง ๆ ไม่ว่าจะเป็นการค้นหาและรวบรวมความต้องการที่ถูกต้องจากผู้ใช้งาน การสร้างข้อกำหนดความต้องการซอฟต์แวร์ให้ตรงตามความต้องการที่ได้รวบรวม การออกแบบและการเขียนโปรแกรมซอฟต์แวร์ให้สามารถทำงานได้ตรงตามข้อกำหนดความต้องการซอฟต์แวร์ที่ได้กำหนด และขั้นตอนหนึ่งที่สำคัญในการพัฒนาซอฟต์แวร์ เพื่อให้ซอฟต์แวร์ที่พัฒนาขึ้นนั้นมีประสิทธิภาพ น่าเชื่อถือ และถูกต้องตามความต้องการของผู้ใช้งาน คือ ขั้นตอนการทดสอบ (Testing)

2.2.1 ระดับของการทดสอบ (John Watkins, 2001)

ในการทดสอบซอฟต์แวร์นั้นสามารถแบ่งระดับการทดสอบซอฟต์แวร์ได้เป็นระดับต่าง ๆ ดังนี้คือ

ทดสอบระดับหน่วย (Unit testing)

เป็นการทดสอบการทำงานของแต่ละ โมดูล (Module) ภายในโปรแกรม กรณีทดสอบจะสร้างขึ้นเพื่อทดสอบการทำงานของโมดูลนั้น โดยแต่ละโมดูลถูกทดสอบแยกจากกัน ทำให้ในบางโมดูลจำเป็นต้องใช้โปรแกรมที่ถูกเขียนขึ้นมาเฉพาะ นั่นคือตัวขับ (Driver) และตัวดำเนินการ (Stub)

โดยตัวขับเป็นโปรแกรมที่ใช้จำลองการเรียกโมดูลที่ต้องการทดสอบพร้อมกับให้ข้อมูลเข้า (Input) และรับผลลัพธ์ (Output) จากการทำงานของโมดูลนั้นและตัวดำเนินการเป็นโปรแกรมจำลองที่โมดูลที่จะทดสอบนั้นต้องการเรียกใช้ ซึ่งกรณีทดสอบของการทดสอบระดับหน่วยต้องระบุตัวขับ และตัวดำเนินการ สำหรับแต่ละโมดูลไว้

การทดสอบการรวม (Integration testing)

เป็นการตรวจสอบข้อผิดพลาดที่เกิดจากการรวมโมดูลเข้าด้วยกัน และเป็นการตรวจสอบความเข้ากันได้ระหว่างโมดูล โดยมีวิธีการในการรวม ดังนี้

- การรวมจากบนลงล่าง (Top-down approach) เป็นการรวมโมดูลที่อยู่ระดับบนสุดในแผนภาพโครงสร้าง (Structure chart) ก่อน ทำให้ต้องมีการสร้างตัวดำเนินการเพื่อให้โมดูลที่รวมเข้าด้วยกันสามารถเรียกใช้ได้ ซึ่งในกรณีทดสอบต้องระบุตัวดำเนินการที่ส่วนที่ทดสอบนั้นเรียกไว้ด้วย
- การรวมจากล่างขึ้นบน (Bottom-up approach) เป็นการรวมโมดูลที่อยู่ระดับล่างสุดในแผนภาพโครงสร้างก่อน ทำให้ต้องมีการสร้างตัวขับเพื่อเรียกและส่ง

ข้อมูลไปยังโมดูลที่รวมเข้าด้วยกัน ในกรณีทดสอบต้องระบุตัวจับ ที่เรียกส่วนที่ทดสอบนั้นด้วย

- การรวมแบบแซนด์วิช (Sandwich approach) เป็นการรวมโมดูลที่อยู่ด้านบนด้วยวิธีการรวมจากบนลงล่าง และรวมโมดูลที่อยู่ด้านล่างด้วยวิธีการแบบล่างขึ้นบน จากนั้นจึงรวมทั้ง 2 ส่วนเข้าด้วยกัน ต้องมีการสร้างทั้งตัวจับ และตัวดำเนินการ เพื่อให้โมดูลที่รวมกันนั้นเรียกใช้
- การรวมแบบโมดูลวิกฤต (Critical module approach) โดยเริ่มจากการรวมโมดูลที่มีความสำคัญมากกว่าโมดูลอื่น ๆ ก่อน เพื่อให้โมดูลเหล่านั้นได้ถูกทดสอบมากขึ้น
- การรวมแบบบิกแบง (Big-bang approach) เป็นการรวมโมดูลเข้าด้วยกันในครั้งเดียว ซึ่งเป็นวิธีที่นิยมใช้ในกรณีที่มียุทธศาสตร์ในการทดสอบจำกัด

การทดสอบระบบ (System testing)

เป็นการตรวจสอบความถูกต้องในการทำงานของส่วน (Element) ต่าง ๆ ในระบบทั้งหมด โดยกรณีทดสอบที่สร้างขึ้นต้องครอบคลุมทุก ๆ หน้าที่ (Function) การทำงานของระบบทั้งซอฟต์แวร์ และฮาร์ดแวร์ เพื่อให้ระบบที่พัฒนาไม่เกิดความผิดพลาดขึ้นในการทำงาน

- การทดสอบตามหน้าที่ (Functional) โดยพยายามทดสอบความต้องการของผู้ใช้ทั้งหมดที่ได้กำหนดไว้ในข้อกำหนดความต้องการซอฟต์แวร์ (Software requirement specification)
- การทดสอบอื่น ๆ (Non functional) ประกอบด้วย
 - o การทดสอบความกดดัน (Stress testing) โดยทดสอบระบบภายใต้สภาพการใช้งานที่สูงมาก ๆ มีจุดประสงค์เพื่อต้องการทราบความสามารถสูงสุดของระบบภายใต้ทรัพยากรที่กำหนด เช่น การทดสอบจำนวนรายการที่เกิดขึ้น (Transaction) มากที่สุดที่ระบบสามารถรองรับได้ เป็นต้น
 - o การทดสอบทางด้านประสิทธิภาพ (Performance testing) เพื่อแสดงว่าระบบมีประสิทธิภาพตามที่กำหนดไว้หรือไม่ เพื่อปรับแต่งระบบให้มีประสิทธิภาพดีขึ้น โดยทำการตรวจสอบจากเวลาตอบสนอง (Response time) ปริมาณงาน (Throughput) การใช้งานหน่วยประมวลผลกลาง (CPU utilization) เป็นต้น
 - o การทดสอบส่วนหลัง (Background testing) เพื่อแสดงความสามารถของระบบในกรณีที่รองรับรายการมากกว่าหนึ่งรายการในเวลาเดียวกัน

- การทดสอบโครงแบบ (Configuration testing) เพื่อต้องการทราบข้อกำหนดของฮาร์ดแวร์ (Hardware) ที่เหมาะสมกับระบบ
- การทดสอบการกู้ระบบ (Recovery testing) เพื่อแสดงว่าระบบสามารถกู้ข้อมูลกลับคืนมาได้หากระบบเกิดความเสียหายขึ้น
- การทดสอบความปลอดภัย (Security testing) เพื่อแสดงว่าระบบที่มีอยู่มีประสิทธิภาพเพียงพอในการรักษาความปลอดภัย และป้องกันระบบจากการลักลอบใช้งานของผู้ที่ไม่ได้รับอนุญาต

การทดสอบย้อนกลับ (Regression testing)

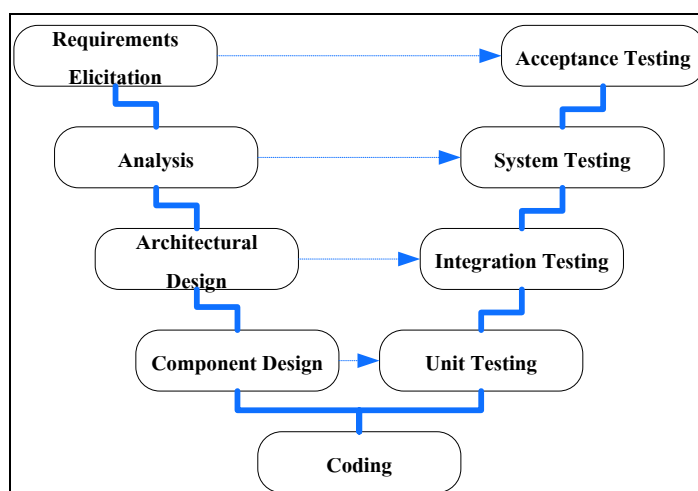
เป็นการทดสอบที่สำคัญหลังจากการแก้ไขซอฟต์แวร์ โดยแบ่งเป็น 2 ประเภทคือ

- การทดสอบหลังจากที่แก้ไขข้อผิดพลาดเรียบร้อยแล้ว เพื่อตรวจสอบการทำงานของซอฟต์แวร์ที่ได้ทำการแก้ไข ว่าทำงานถูกต้องตามที่ได้ออกแบบไว้หรือไม่
- การทดสอบหลังจากที่มีการเปลี่ยนแปลงหรือแก้ไขโปรแกรม เพื่อให้มั่นใจได้ว่าการทำงานของแต่ละโมดูลในโปรแกรมยังคงไม่มีข้อผิดพลาด และถูกต้องตามที่ได้ออกแบบไว้

การทดสอบการยอมรับ (Acceptance testing)

วัตถุประสงค์ของการทดสอบการยอมรับ เพื่อให้ยืนยันว่าซอฟต์แวร์ที่ทดสอบนั้นสามารถทำงานได้ตรงตามความต้องการของผู้ใช้งานและสร้างความเชื่อมั่นว่าระบบสามารถทำงานได้ถูกต้องก่อนที่จะส่งมอบอย่างเป็นทางการให้แก่ผู้ใช้งาน การทดสอบการยอมรับ ถูกดำเนินโดยตัวแทนของผู้ใช้งาน อาจจะเป็นคนเดียวหรือมากกว่าก็ได้ ซึ่งจะได้รับความช่วยเหลือจากทีมงานของผู้ทดสอบ การทดสอบการยอมรับใช้เทคนิคแบบแบล็กบ็อกซ์ (Black-box testing) โดยใช้เทคนิคการสร้าง Thread testing เป็นลักษณะการทดสอบตามงานที่ทำในลักษณะของการใช้งานระบบ เพื่อตรวจสอบความต้องการทางด้านธุรกิจในระดับสูง (High-level business requirement) นอกจากนี้ การทดสอบการยอมรับจะต้องดูจุดที่ต้องทดสอบตามเอกสารของระบบ โดยตัวแทนผู้ใช้งาน การทดสอบการยอมรับจะสัมพันธ์กับช่วงของขั้นตอนการเก็บรวบรวมความต้องการของ V-Model

V-Model เป็นกระบวนการของการพัฒนาและทดสอบซอฟต์แวร์ และที่สำคัญคือการวางแผนและเตรียมการทดสอบ ตั้งแต่ขั้นตอนการพัฒนา ซึ่งในฝั่งซ้ายของ V-Model จะเป็นขั้นตอนการพัฒนาซอฟต์แวร์แบบน้ำตก (Waterfall Model) และทางด้านขวาจะเป็นขั้นตอนของการทดสอบ



รูปที่ 2.6 V-Model

การทดสอบการยอมรับจะสัมพันธ์ กับช่วงของ Requirements Phase ของ V-Model ซึ่ง Phase ของการทำ Requirements Elicitation นั้นจะต้องมีการวางแผนการทบทวนเพื่อตรวจสอบความต้องการ และกำหนดตัวแทนของผู้ใช้งานไว้ล่วงหน้าก่อนทำการทดสอบการยอมรับ

2.2.2 กรณีทดสอบ (Test case)

กรณีทดสอบ คือข้อมูลที่ใช้ทดสอบใช้ทำการทดสอบรายการ (Item) ใด ๆ เพื่อให้ทราบว่าการทำงานของรายการนั้นถูกต้องหรือไม่ ประกอบด้วยข้อมูลเข้า (Input) ผลที่คาดว่าจะได้ (Expected output) และข้อจำกัดต่าง ๆ ในการใช้กรณีทดสอบนั้น กรณีทดสอบที่ดีต้องมีคุณสมบัติ คือ มีความน่าจะเป็นสูงที่จะเจอข้อผิดพลาด (Error) ซึ่งยังไม่ถูกค้นพบในโปรแกรม และกรณีทดสอบที่ประสบความสำเร็จ คือกรณีทดสอบที่สามารถค้นพบข้อผิดพลาดที่เกิดขึ้นในโปรแกรมได้

การเลือกกรณีทดสอบเพื่อนำไปทำการทดสอบ มีหลักการในการพิจารณาดังนี้

- กรณีทดสอบที่ใช้ทดสอบความสามารถ ความมั่นคงของระบบมีความสำคัญมากกว่ากรณีทดสอบที่ใช้ทดสอบส่วนประกอบของระบบ เนื่องจากผู้ใช้จะให้ความสำคัญกับความมั่นคงของระบบในการทำงานมากกว่าข้อผิดพลาดเล็กน้อยที่เกิดขึ้นในระบบ ดังเช่นข้อผิดพลาดที่ทำให้เกิดการสูญเสีย หรือสูญหาย ของข้อมูล มีความสำคัญมากกว่าข้อผิดพลาดที่ทำให้ข้อมูลที่แสดงทางจอภาพขาดหายไป

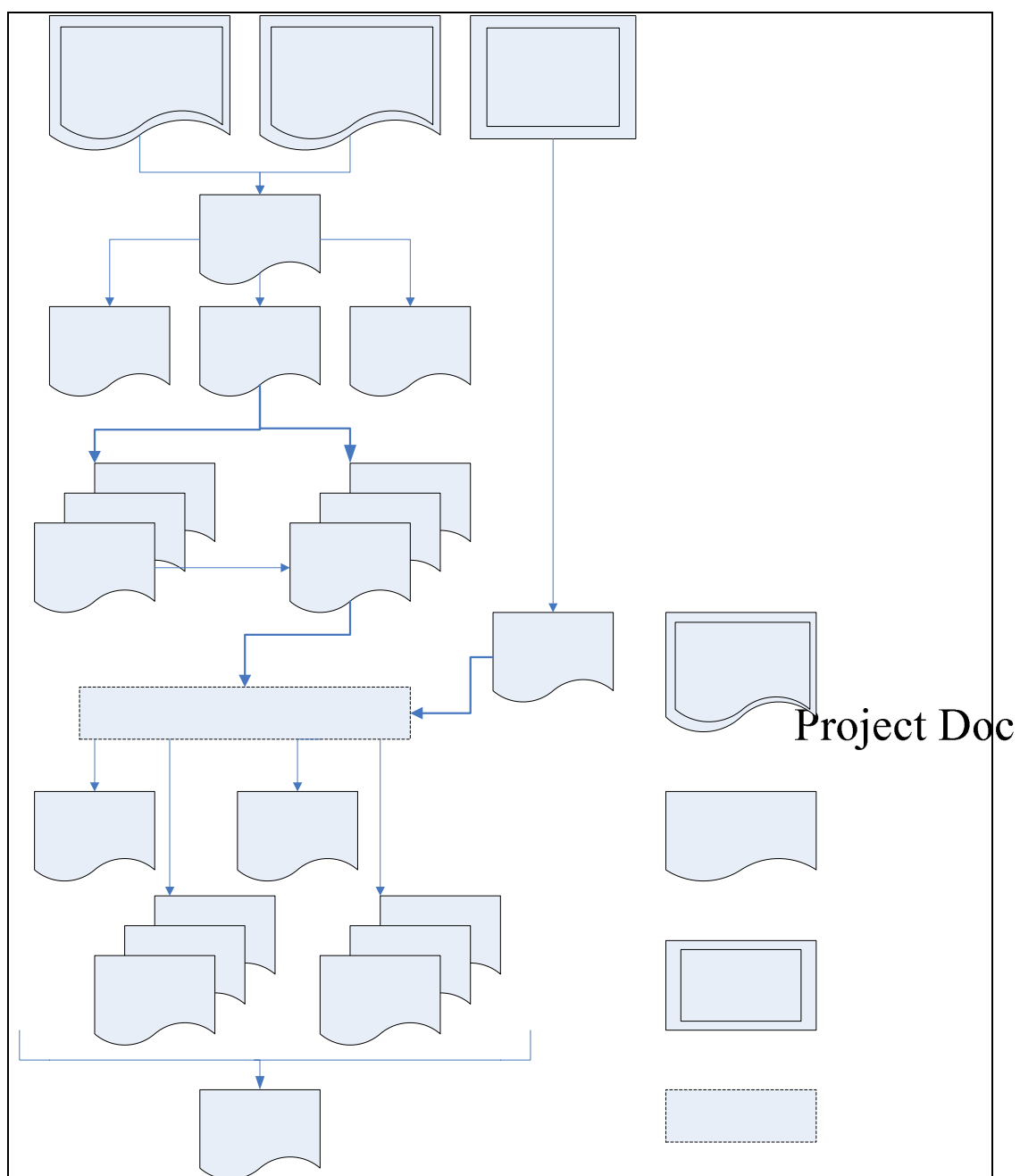
- ในกรณีที่มีการปรับปรุง หรือแก้ไขระบบใหม่ กรณีทดสอบที่ใช้ทดสอบความสามารถเดิมของระบบ มีความสำคัญมากกว่ากรณีทดสอบสำหรับใช้ทดสอบความสามารถของระบบที่พัฒนาขึ้นใหม่
- ในกรณีที่มีข้อจำกัดในเรื่องของจำนวนกรณีทดสอบ กรณีทดสอบที่มีข้อมูลทดสอบในลักษณะทั่วไปมีความสำคัญมากกว่ากรณีทดสอบที่มีข้อมูลทดสอบอยู่ในช่วงที่เป็นค่าขอบเขต

การออกแบบกรณีทดสอบในแต่ละระดับแบ่งออกเป็นสองประเภท คือ

- การออกแบบกรณีทดสอบแบบไวท์บ็อกซ์ (White-box testing) เป็นการสร้างกรณีทดสอบโดยพิจารณาจากโครงสร้าง (Structure) ของโปรแกรม โดยสร้างกรณีทดสอบที่ครอบคลุมทุกเส้นทาง (Path) ตามโครงสร้างของโปรแกรมเพื่อให้แน่ใจว่าทุกคำสั่งในโปรแกรมได้รับการทดสอบอย่างน้อยหนึ่งครั้ง และทุกคำสั่งที่เป็นเงื่อนไขถูกทดสอบ ทั้งกรณีที่เงื่อนไขนั้นเป็นจริงและกรณีที่เงื่อนไขนั้นเป็นเท็จ นอกจากนี้ยังสามารถค้นพบข้อผิดพลาดที่เกิดขึ้นจากข้อมูลภายในโมดูลได้อีกด้วย การสร้างกรณีทดสอบด้วยวิธีการของไวท์บ็อกซ์ ผู้ทดสอบต้องสร้างกราฟการควบคุมกระแส (Control flow graph) จากโปรแกรมที่พัฒนาขึ้น และนำกราฟที่ได้มาสร้างกรณีทดสอบเพื่อให้ครอบคลุมทุกคำสั่งทุกเส้นทาง และทุกเงื่อนไขของโปรแกรม
- การออกแบบกรณีทดสอบแบบแบล็กบ็อกซ์ (Black-box testing) เป็นการสร้างกรณีทดสอบโดยพิจารณาจากข้อกำหนด (Specification) และเงื่อนไข (Condition) ของข้อมูลนั้น การทดสอบสนใจเฉพาะข้อมูลเข้าและผลลัพธ์ของโมดูลที่ต้องการทดสอบ เพื่อให้ตรงตามข้อกำหนดที่ได้ออกแบบไว้เท่านั้น ซึ่งจะไม่พิจารณาถึงโครงสร้างภายในโมดูลนั้น

2.2.3 มาตรฐาน IEEE Std 829-1998

เป็นมาตรฐานที่เกี่ยวกับเอกสาร และข้อมูลที่ใช้ในการทดสอบ โดยมาตรฐานนี้อธิบายเกี่ยวกับจุดมุ่งหมาย แนวทาง และองค์ประกอบเบื้องต้นของเอกสารและข้อมูลที่ใช้ในการทดสอบซอฟต์แวร์ ดังแสดงในรูปที่ 2.3 เพื่อให้เป็นมาตรฐานและเกิดความเข้าใจตรงกันระหว่างผู้พัฒนาและลูกค้า อีกทั้งยังทำให้การทดสอบควบคุมได้ เนื่องจากสามารถตรวจสอบการทำงานในแต่ละขั้นตอนจากเอกสารและ ข้อมูลที่บันทึกไว้ได้ ซึ่งมาตรฐานนี้ถูกออกแบบมาโดยไม่ขึ้นกับประเภทซอฟต์แวร์ ขนาดของคอมพิวเตอร์ วิธีการทดสอบ หรือเครื่องมือที่ช่วยในการทดสอบต่างๆ สามารถนำมาใช้ได้ในการทดสอบซอฟต์แวร์ทุกประเภท และในทุกระดับของการทดสอบ มาตรฐานนี้ประกอบด้วยส่วนต่างๆ ดังต่อไปนี้



รูปที่ 2.7 ความสัมพันธ์ของเอกสารที่ใช้ในการทดสอบตามมาตรฐาน IEEE Std.829-1998

Test Design Spec

Test

การวางแผนการทดสอบ (Test Planning) ระบุถึงขอบเขต แนวทาง ทรัพยากร และ กำหนดการ ของการทดสอบ ระบุส่วนของโปรแกรมลักษณะ (Feature) ที่จะทดสอบ กิจกรรมที่ต้อง ทำ ความรับผิดชอบของผู้ทดสอบ และความเสี่ยงที่จะเกิดขึ้นในแผนการทดสอบ

รายละเอียดของการทดสอบ (Test Specification) ประกอบด้วยเอกสารและข้อมูล ต่าง ๆ ได้แก่

- 1) รายละเอียดเกี่ยวกับการออกแบบการทดสอบ (Test-design Specification) ระบุ การออกแบบการทดสอบ เพื่ออธิบายลักษณะที่จะทดสอบที่ได้กำหนดไว้ใน การวางแผนการทดสอบ นอกจากนั้นยังระบุถึงกรณีทดสอบที่เกี่ยวข้องและ หลักในการพิจารณาการทดสอบในแต่ละกรณีทดสอบว่าผ่านหรือไม่ผ่าน เกณฑ์ที่กำหนด
- 2) รายละเอียดเกี่ยวกับกรณีทดสอบ (Test-case specification) ระบุค่าของข้อมูล เข้าเพื่อใช้ในการทดสอบและกำหนดผลลัพธ์ที่ต้องการ นอกจากนั้นยังระบุ ข้อจำกัดต่าง ๆ ที่มีในการใช้กรณีทดสอบนั้น ข้อมูลในรายละเอียดเกี่ยวกับ กรณีทดสอบ ได้แก่
 - ตัวระบุกรณีทดสอบ (Test case specification identifier) เพื่อให้ผู้ทดสอบ สามารถระบุกรณีทดสอบที่ใช้ทดสอบแต่ละรายการทดสอบได้
 - รายการที่ทดสอบ (Test item) แสดงชื่อ และ/หรือ หน้าที่ของรายการที่จะ ทดสอบสำหรับกรณีทดสอบนั้น
 - ค่าของข้อมูลเข้าเพื่อใช้ในการทดสอบ (Input specification)
 - ผลลัพธ์ที่คาดว่าจะได้รับจากการทดสอบ (Output specification)
 - ลักษณะสิ่งแวดล้อมที่ต้องการในการทดสอบ (Environment needs) เช่น ลักษณะอุปกรณ์ ซอฟต์แวร์
 - ความต้องการพิเศษอื่น ๆ (Special procedural requirement) ระบุความ ต้องการพิเศษอื่น ๆ เพื่อใช้สำหรับกรณีทดสอบนี้
 - รายการกรณีทดสอบที่เกี่ยวข้อง (Intercase dependency) แสดงรายการกรณี ทดสอบที่ต้องทำการทดสอบก่อนหน้ากรณีทดสอบนี้ และความสัมพันธ์กับ กรณีทดสอบนี้
- 3) ขั้นตอนการทดสอบ (Test procedure specification) ระบุขั้นตอนทั้งหมดที่ต้อง ทำในการจัดการทดสอบโดยใช้กรณีทดสอบที่ได้ออกแบบไว้ วัตถุประสงค์ ของขั้นตอนการทดสอบ และความต้องการพิเศษที่ใช้ในการทดสอบกรณี ทดสอบตามขั้นตอนที่ระบุ

- 4) การรายงานผลการทดสอบ (Test reporting) ประกอบด้วยเอกสารต่าง ๆ ดังนี้
 - รายงานส่วนของโปรแกรมที่ถูกส่งไปทดสอบ (Test item transmitted report) ระบุรายละเอียดของรายการทดสอบ สถานะของรายการทดสอบ ผู้รับผิดชอบการทดสอบ
 - บันทึกการทดสอบ (Test log) บันทึกผลลัพธ์ที่ได้จากการทดสอบโดยใช้กรณีทดสอบต่าง ๆ รวมทั้งกิจกรรมและเหตุการณ์ที่เกิดขึ้น (Activity and event entries) ประกอบด้วยข้อมูลดังนี้
 - คำอธิบายการกระทำ (Execution description) อธิบายลักษณะ ขั้นตอนการทดสอบ ผู้ที่เกี่ยวข้องกับการทดสอบ วัน เวลาที่ทดสอบ และเหตุการณ์ต่างที่เกิดขึ้นในขณะทดสอบ
 - ผลการทดสอบแต่ละขั้นตอน (Procedure result) ผลการทดสอบ และผลลัพธ์ (Output) ที่ได้จากการทดสอบแต่ละขั้นตอน
 - ข้อมูลสภาพแวดล้อม (Environment information) ระบุสภาพแวดล้อม อุปกรณ์ ซอฟต์แวร์ต่าง ๆ ในขณะทำการทดสอบ
 - เหตุการณ์ผิดปกติ (Anomalous event) ระบุลักษณะเหตุการณ์ผิดปกติที่เกิดขึ้น เหตุการณ์ก่อนและหลัง และสภาพแวดล้อมก่อน และหลังเกิดเหตุผิดปกติ
- 5) รายงานการทดสอบที่ต้องดำเนินการต่อ (Test incident report) ระบุถึงเหตุการณ์ที่เกิดขึ้นระหว่างการทดสอบซึ่งต้องการการติดตามตรวจสอบต่อ ประกอบด้วย
 - ผลสรุปเหตุการณ์ที่เกิดขึ้น (Summary) สรุปเหตุการณ์ที่เกิดขึ้น ลำดับขั้นตอนการเกิดเหตุการณ์ รายการที่ทดสอบ ลำดับขั้นตอนการทดสอบ
 - คำอธิบายเหตุการณ์ (Incident description) ลักษณะเหตุการณ์ที่เกิดขึ้น วันที่ เวลาที่เกิด เหตุผิดปกติที่เกิด ข้อมูลเข้า ผลลัพธ์ ผลที่คาดว่าจะได้รับ และผู้ทดสอบ
 - ผลกระทบของเหตุการณ์ที่เกิดขึ้น (Impact) ผลกระทบของเหตุการณ์ที่เกิดขึ้นกับระบบ หรือ กับกรณีทดสอบอื่น ๆ
- 6) รายงานสรุปผลการทดสอบ (Test summary report) สรุปกิจกรรมที่เกิดขึ้นตามที่ได้ระบุไว้ในรายละเอียดเกี่ยวกับการออกแบบการทดสอบ สรุปผลการทดสอบ การประเมินผลการทดสอบ และผู้ที่รับผิดชอบการทดสอบ

2.3 งานวิจัยที่เกี่ยวข้อง

2.3.1 งานวิจัยเกี่ยวกับการประเมินผลมาตรฐานเอกสารข้อกำหนดความต้องการซอฟต์แวร์ทั้ง 4 มาตรฐาน

(Giakoumakis and Xylomenos, 1996) Giakoumakis และ Xylomenos ทำการประเมินมาตรฐานเอกสารข้อกำหนดความต้องการซอฟต์แวร์ทั้ง 4 มาตรฐาน ได้แก่ มาตรฐาน IEEE Std 830-1933, มาตรฐาน ESA PSS-05-0, มาตรฐาน NASA-DID-P200 และมาตรฐาน DoD-STD-498 โดยมีการสร้างเกณฑ์และมีการประเมินผลมาตรฐาน 8 ข้อ คือ

- 1) ความเป็นอิสระ (Independence) พิจารณาจากระเบียบวิธีที่ใช้ในการพัฒนาซอฟต์แวร์ ผู้พัฒนาซอฟต์แวร์สามารถใช้ระเบียบวิธีที่เป็นอิสระได้ เพราะเป็นการหลีกเลี่ยงความไม่แน่นอนของมาตรฐาน ซึ่งการประเมินผลแบ่งเป็น 3 ระดับ คือ
 - ระดับสูง มาตรฐานมีรูปแบบวัฏจักรของการพัฒนาซอฟต์แวร์หลายรูปแบบที่ใช้ในการออกแบบระบบ และประยุกต์ใช้กับระเบียบวิธีของการพัฒนาซอฟต์แวร์ที่แตกต่างกันได้
 - ระดับกลาง มาตรฐานบางมาตรฐานไม่สามารถประยุกต์ใช้กับวัฏจักรของการพัฒนาซอฟต์แวร์หลายรูปแบบ
 - ระดับต่ำ มาตรฐานบางมาตรฐานได้รับการพิจารณาว่าไม่เหมาะสมที่จะนำมาใช้ในการพัฒนาซอฟต์แวร์
- 2) การรวมเป็นหนึ่งเดียว (Integration) พิจารณาจากความสำเร็จของมาตรฐานเอกสารข้อกำหนดความต้องการซอฟต์แวร์ว่าครอบคลุมขั้นตอนของการพัฒนาซอฟต์แวร์ทั้งหมดหรือไม่ ซึ่งการประเมินผล แบ่งเป็น 3 ระดับ คือ
 - ระดับสูง มาตรฐานมีความสมบูรณ์ สามารถใช้มาตรฐานเพื่อเปลี่ยนแปลงแก้ไขซอฟต์แวร์ได้ง่าย
 - ระดับกลาง มีบางส่วนของมาตรฐานที่ไม่สมบูรณ์ ไม่ครอบคลุมขั้นตอนบางขั้นตอนของการพัฒนาซอฟต์แวร์
 - ระดับต่ำ มาตรฐานมีความไม่สมบูรณ์ ไม่ครอบคลุมขั้นตอนของการพัฒนาซอฟต์แวร์
- 3) ความถูกต้อง (Precision) พิจารณาจากความกะทัดรัดและความชัดเจนของเนื้อหา ต้องมีความเข้าใจที่ตรงกัน หรือไม่มีความคลุมเครือเกิดขึ้น พร้อมทั้งมีความยืดหยุ่น ซึ่งการประเมินผล แบ่งเป็น 3 ระดับ คือ

- ระดับสูง รายละเอียดในความต้องการมีความสมบูรณ์ ชัดเจน และมีตัวอย่างที่ละเอียดในแต่ละหัวข้อ
 - ระดับกลาง มาตรฐานมีความชัดเจน แต่ยังคงมีค่าใช้จ่ายและความเสี่ยงเล็กน้อย
 - ระดับต่ำ มาตรฐานมีความคลุมเครือพร้อมทั้งมีค่าใช้จ่ายและความเสี่ยงมาก
- 4) คุณลักษณะโดยทั่วไป (Generality) พิจารณาจากความเหมาะสมของมาตรฐานที่จะใช้กับโครงการต่าง ๆ ในองค์กร ซึ่งการประเมินผล แบ่งเป็น 2 ลักษณะ คือ
- โครงการขนาดใหญ่
 - โครงการขนาดเล็ก
- 5) การจัดระเบียบของเอกสาร (Organization) พิจารณาจากเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ต้องมีความยืดหยุ่น สามารถเข้าใจได้ง่าย และใช้งานง่าย ซึ่งการประเมินผล แบ่งเป็น 3 ระดับ คือ
- ระดับสูง เอกสารข้อกำหนดความต้องการซอฟต์แวร์มีความยืดหยุ่นสูง
 - ระดับกลาง เอกสารข้อกำหนดความต้องการซอฟต์แวร์ไม่มีความยืดหยุ่น มีโครงสร้างเป็นแบบคงที่
 - ระดับต่ำ เอกสารข้อกำหนดความต้องการซอฟต์แวร์ยากต่อการเข้าใจ ทำให้เสียค่าใช้จ่ายสูงในการบำรุงรักษาระบบ
- 6) ความสมบูรณ์ของเนื้อหา (Content completeness) พิจารณาจากความต้องการทุกประเภทต้องอยู่ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ทั้งหมด จุดประสงค์เพื่อใช้เป็นแนวทางในการวิเคราะห์ระบบและเพื่อตรวจสอบซอฟต์แวร์ ซึ่งการประเมิน แบ่งเป็น 3 ระดับ คือ
- ระดับสูง มาตรฐานต้องครอบคลุมความต้องการทั้งหมด รายละเอียดในมาตรฐานไม่จำเป็นต้องละเอียดมาก แต่ควรที่จะใช้ง่าย และไม่คลุมเครือ
 - ระดับกลาง มาตรฐานพยายามที่จะควบคุมความต้องการทั้งหมด เพื่อไม่ให้มีความไม่สอดคล้องกัน และความคลุมเครือเกิดขึ้นในเอกสาร
 - ระดับต่ำ ความต้องการบางอย่างมีความสมบูรณ์ แต่ไม่ครอบคลุมความต้องการทั้งหมดในเอกสารข้อกำหนดความต้องการซอฟต์แวร์
- 7) ความสมบูรณ์ของมุมมองภายนอก (View completeness) พิจารณาจากภาพพจน์ของผู้ใช้งาน นักวิเคราะห์ระบบ นักออกแบบระบบ นักทดสอบระบบ และนักบำรุงรักษาระบบ

- ผู้ใช้งาน ความต้องการต่าง ๆ มาจากผู้ใช้งาน เอกสารข้อกำหนดความต้องการซอฟต์แวร์ต้องได้รับการตรวจสอบว่าถูกต้องและเข้าใจง่ายหรือไม่
 - นักวิเคราะห์ระบบ ทำการรวบรวมความต้องการจากผู้ใช้หรือลูกค้าและเขียนเอกสารข้อกำหนดความต้องการซอฟต์แวร์ให้มีความสมบูรณ์ เพื่อใช้เป็นแนวทางในการวิเคราะห์ระบบ
 - นักออกแบบระบบ ใช้เอกสารข้อกำหนดความต้องการซอฟต์แวร์ในการพัฒนาซอฟต์แวร์
 - นักทดสอบระบบ ทำการตรวจสอบเอกสารข้อกำหนดความต้องการซอฟต์แวร์ว่าตรงกับความต้องการของผู้ใช้หรือลูกค้าหรือไม่
 - นักบำรุงรักษาระบบ ใช้เอกสารข้อกำหนดความต้องการซอฟต์แวร์ เพื่อเปลี่ยนแปลงแก้ไขซอฟต์แวร์ได้ง่าย
- 8) การเปลี่ยนแปลงแก้ไขได้ (Modifiability) พิจารณาจากวิธีการแก้ไขเอกสารข้อกำหนดความต้องการซอฟต์แวร์ โดยไม่ส่งผลกระทบต่อคุณภาพของเอกสารซึ่งการประเมินผล แบ่งเป็น 2 ลักษณะคือ
- เพียงพอ (*adequate*) ผู้พัฒนาซอฟต์แวร์สามารถกระจายความต้องการออกเป็นโมดูล ตลอดจนสามารถมีการติดตามย้อนหลัง และไปข้างหน้าได้
 - ไม่เพียงพอ (*inadequate*) การเปลี่ยนแปลงแก้ไขส่งผลกระทบต่อเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ผู้พัฒนาซอฟต์แวร์ไม่สามารถใช้เอกสารข้อกำหนดความต้องการซอฟต์แวร์ในการติดตามซอฟต์แวร์ได้

งานวิจัยของ Giakoumakis และ Xylomenos ได้สรุปผลการประเมินผลมาตรฐานทั้งสี่ ดังแสดงในตารางที่ 2.2

ตารางที่ 2.3 การประเมินผลของมาตรฐาน IEEE Std.830-1993, ESA PSS-05-0, NASA-DID-P200 และ DOD-STD-498

เกณฑ์	การประเมินผล			
	IEEE Std.830-1993	ESA PSS-05-0	NASA-DID-P200	DOD-STD-498
ความเป็นอิสระ	ปานกลาง	ปานกลาง	สูง	สูง
การรวมเป็นหนึ่งเดียว	ปานกลาง	ปานกลาง	สูง	สูง
ความถูกต้อง	สูง	ต่ำ	สูง	สูง
คุณลักษณะโดยทั่วไป	ไม่เหมาะสมกับโครงการขนาดใหญ่	ไม่เหมาะสมกับโครงการขนาดใหญ่	ไม่เหมาะสมกับโครงการขนาดเล็ก	ไม่เหมาะสมกับโครงการขนาดเล็ก
การจัดระเบียบของเอกสาร	สูง	ต่ำ	ต่ำ	ปานกลางถึงสูง
ความสมบูรณ์ของเนื้อหา	สูง	ปานกลาง	สูง	สูง
ความสมบูรณ์ของมุมมองภายนอก	เหมาะสมกับผู้ใช้ทุกคน	เหมาะสมกับนักทดสอบระบบเท่านั้น	เหมาะสมกับนักออกแบบระบบและนักทดสอบระบบ	เหมาะสมกับผู้ใช้ทุกคน
การเปลี่ยนแปลงแก้ไขได้	เพียงพอ	ไม่เพียงพอ	ไม่เพียงพอ	เพียงพอ

พร้อมทั้งศึกษางานวิจัยเกี่ยวกับการพัฒนาแนวทางสำหรับการเตรียมเอกสารข้อกำหนดความต้องการซอฟต์แวร์ (น้ำฝน อัสวเมธิน, 2542) ซึ่งสรุปผลสำหรับเนื้อหาที่ควรมีในเอกสารข้อกำหนดความต้องการซอฟต์แวร์จำนวน 46 เนื้อหา โดยแบ่งเป็น 4 ส่วน อันได้แก่

- 1) บทนำ ประกอบไปด้วย จุดประสงค์ ขอบเขต คำจำกัดความ รหัสพจน์ คำย่อ เอกสารอ้างอิง และภาพรวม
- 2) รายละเอียดโดยรวม ประกอบไปด้วย การต่อประสานกับผู้ใช้ การต่อประสานกับฮาร์ดแวร์ การต่อประสานกับซอฟต์แวร์ การต่อประสานกับระบบสื่อสาร ข้อกำหนดของหน่วยความจำ การดำเนินงาน ความต้องการด้านการติดตั้งซอฟต์แวร์ หน้าที่ของซอฟต์แวร์ ข้อกำหนดในการเตรียมกระบวนการในการพิจารณาซอฟต์แวร์ ความสามารถในการติดตามความต้องการ ข้อกำหนด ข้อสันนิษฐานและการขึ้นต่อกัน และการแบ่งส่วนปันส่วนความต้องการ
- 3) ความต้องการที่เฉพาะเจาะจง ประกอบไปด้วย การต่อประสานภายนอก ความต้องการด้านหน้าที่ ความต้องการด้านประสิทธิภาพ ความต้องการด้านฐานข้อมูลเชิงตรรกะ ความต้องการด้านการดำเนินการ ความต้องการด้านการทวนสอบ ความต้องการด้านการยอมรับ ความต้องการด้านเอกสาร ความต้องการด้านความปลอดภัย ความต้องการด้านการใช้ทรัพยากรฮาร์ดแวร์ คอมพิวเตอร์ให้เกิดประโยชน์สูงสุด ความต้องการด้านระบบสื่อสาร

คอมพิวเตอร์ ปัจจัยที่มีผลต่อคุณภาพของซอฟต์แวร์ ความต้องการด้านการฝึกอบรมบุคคลที่เกี่ยวข้อง ความต้องการด้านการเตรียมขั้นตอนต่าง ๆ ที่ใช้ในการพัฒนาซอฟต์แวร์ ความต้องการด้านบรรจุภัณฑ์ซอฟต์แวร์ ข้อจำกัดในการใช้งานซอฟต์แวร์ มาตรฐานที่มีในองค์กร ความเชื่อถือได้ สภาพพร้อมใช้งาน ความมั่นคง ความสามารถในการบำรุงรักษา และความสามารถในการใช้ได้หลายระบบ

4) ข้อมูลสนับสนุน ประกอบไปด้วย สารบัญ บันทึก ภาคผนวก และดัชนี

2.3.2 ระบบจัดการกรณีทดสอบ (Test case management system) (Hiren, 1994)

พัฒนาโดย Hiren D. Desai เป็นระบบที่พัฒนาขึ้นเพื่อใช้ในการทดสอบโปรแกรมในหน่วยงานของบริษัทเบลเซาท์ (BellSouth Co. Ltd.) ที่ประเทศสหรัฐอเมริกา ระบบจะเก็บรวบรวมข้อมูลกรณีทดสอบและผลการทดสอบสำหรับโครงการต่าง ๆ ที่กำลังพัฒนา ผู้ทดสอบจะทำการบันทึกกรณีทดสอบที่ได้ออกแบบไว้ หลังจากนั้นระบบจะสร้างบทคำสั่งการทดสอบ (Test script) เพื่อนำไปทดสอบโปรแกรม ในขั้นตอนการทดสอบโปรแกรมระบบสามารถรับผลการทดสอบจากผู้ทดสอบ หรือทดสอบและบันทึกผลการทดสอบโปรแกรมได้โดยอัตโนมัติ เครื่องมือนี้สามารถนำข้อมูลผลการทดสอบมาประมวลผล และออกรายงานสรุปผลการทดสอบ และรายงานความก้าวหน้าในการทดสอบให้กับผู้ใช้ นอกจากนี้ผู้ใช้งานยังสามารถกำหนดรูปแบบ และข้อมูลที่จะแสดงในรายงานได้ อีกด้วยระบบนี้สามารถทดสอบโปรแกรมที่เขียนด้วยภาษาซี (C language) และทำงานบนระบบปฏิบัติการยูนิกซ์ (Unix) ซึ่งทำให้การพัฒนาโปรแกรมโดยใช้ภาษาอื่นไม่สามารถทำงานแบบอัตโนมัติได้

2.3.3 TestDirector 6 (Mercury Interactive Ltd, 2000)

ถูกพัฒนาโดย บริษัท Mercury Interactive เป็นเครื่องมือที่ช่วยองค์กรในการทดสอบซอฟต์แวร์ทุกขั้นตอน โดยเริ่มจากการวางแผนการทดสอบ การดำเนินการทดสอบ และการจัดการข้อผิดพลาดที่เกิดขึ้น ในขั้นตอนการวางแผนการทดสอบ เครื่องมือนี้จะช่วยสร้างแผนการทดสอบโดยผู้ใช้งานสามารถสร้างแผนการทดสอบในไมโครซอฟต์เวิร์ด (Microsoft Word) และนำเข้ามาใช้ในเครื่องมือนี้ได้ทันที จากนั้นผู้ทดสอบจะเป็นผู้สร้าง และบันทึกกรณีทดสอบเข้าสู่ระบบ ในส่วนของการดำเนินการทดสอบ เครื่องมื่อดังกล่าวสามารถทำการทดสอบโดยอัตโนมัติ หรือให้ผู้ทดสอบดำเนินการทดสอบเองแล้วรายงานผลการทดสอบกลับมายังเครื่องมือ หากเกิดข้อผิดพลาด เครื่องมือสามารถให้ข้อมูลที่เป็นประโยชน์ ในการวิเคราะห์ข้อผิดพลาดเพื่อหาแนวทางแก้ไข นอกจากนี้เครื่องมือยังสามารถออกรายงานสรุปผลการทดสอบตามรูปแบบที่ใช้กำหนด โดยส่งข้อมูลไปยังไมโครซอฟต์เอ็กเซล (Microsoft Excel) และไมโครซอฟต์เวิร์ด (Microsoft Word) ได้อีกด้วย

2.3.4 พลัสวันเทส (+1 TEST) (+1 Software Engineering, 2002)

เป็นระบบจัดการกรณีทดสอบของบริษัทพลัสวันซอฟต์แวร์เอ็นจิเนียริ่ง (+1 Software Engineering Co. Ltd.) ซึ่งสามารถใช้ได้กับทุกขั้นตอนในการทดสอบโปรแกรมทั้งการทดสอบระดับหน่วย การทดสอบการรวม การทดสอบย้อนกลับ และการทดสอบระบบ โดยระบบจะสร้างบทคำสั่งการทดสอบ เพื่อให้ผู้ทดสอบนำไปทดสอบกลับโปรแกรมที่ถูกพัฒนาขึ้น และนำผลการทดสอบมาทำการบันทึก เข้าสู่ระบบเพื่อที่ระบบจะนำผลการทดสอบมาสรุป และออกรายงานผลการทดสอบต่อไป ผู้ใช้สามารถเพิ่ม-ลด หรือแก้ไขข้อมูลกรณีทดสอบที่ระบบบันทึกไว้ได้ ระบบพลัสวันเทสสามารถสร้างบททดสอบโปรแกรมที่พัฒนาขึ้นด้วยภาษาซี และทำงานบนเครื่องซันเวิร์กสเตชัน (SUN Workstation) บนระบบปฏิบัติการโซลาริส (Solaris 1.x and 2.x)

บทที่ 3

ระเบียบวิธีวิจัย

งานวิจัยนี้มีจุดมุ่งหมายในการออกแบบและพัฒนาเครื่องมือช่วยในการสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ เพื่อนำมาช่วยผู้ทดสอบในการสร้างกรณีทดสอบให้ครอบคลุมเนื้อหาที่ระบุไว้ในข้อกำหนดความต้องการซอฟต์แวร์ และนอกจากนั้น ยังช่วยในการจัดเก็บข้อมูลกรณีทดสอบที่ใช้ในการทดสอบลงฐานข้อมูล เพื่อประโยชน์ในการนำกลับมาใช้ใหม่ และเป็นการลดภาระของผู้ทดสอบอีกด้วย โดยในงานวิจัยนี้จะศึกษาถึงวิธีการในการจัดเก็บความต้องการเชิงหน้าที่ ที่ได้มาจากข้อกำหนดความต้องการซอฟต์แวร์ซึ่งอ้างอิงมาจากมาตรฐานต่างประเทศ 4 มาตรฐาน ได้แก่ มาตรฐาน IEEE Std.830-1993 มาตรฐาน ESA PSS-05-0 มาตรฐาน NASA-DID-P200 และมาตรฐาน DOD-STD-498 และนำเอาความต้องการเชิงหน้าที่ ที่ได้มาจากข้อกำหนดความต้องการซอฟต์แวร์ มาสร้างเป็นกรณีทดสอบในส่วนของการทดสอบการยอมรับ โดยกรณีทดสอบที่ได้จะครอบคลุมและตรงตามความต้องการของผู้ใช้งานมากขึ้น รายละเอียดในเนื้อหาบทนี้ประกอบด้วยระเบียบวิธีวิจัย ปรากฏอยู่ในหัวข้อ 3.1 การวิเคราะห์และออกแบบเครื่องมือช่วยในการสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ ปรากฏอยู่ในหัวข้อที่ 3.2 การออกแบบและสร้างฐานข้อมูล สำหรับเครื่องมือช่วยในการสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ ปรากฏอยู่ในหัวข้อ 3.3 และหัวข้อ 3.4 เป็นการออกแบบส่วนนำเข้าข้อมูล

3.1 ระเบียบวิธีวิจัย

การค้นคว้าวิจัยจะแบ่งออกเป็น ขั้นตอนดังนี้

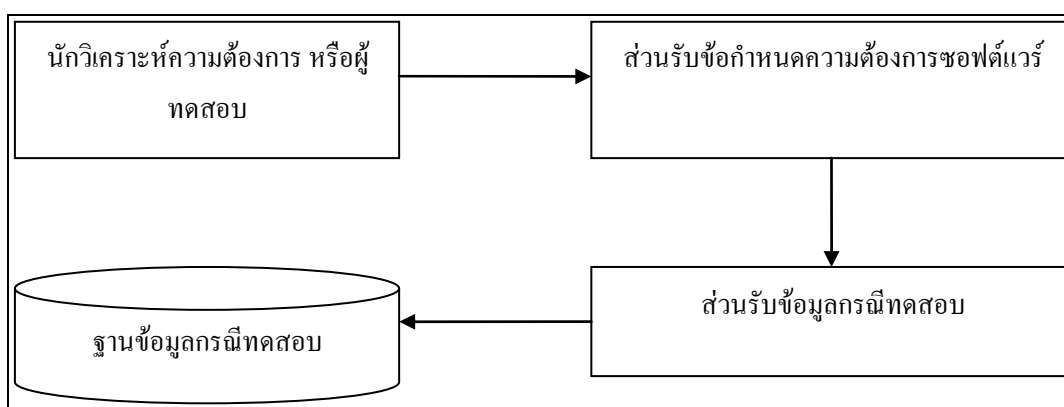
- 3.1.1 ศึกษามาตรฐานของเอกสารที่ใช้ในการกำหนดความต้องการซอฟต์แวร์ ที่มีใช้ใน
เมืองไทย
- 3.1.2 ศึกษากระบวนการจัดการกรณีทดสอบและเครื่องมือที่มีอยู่ในปัจจุบัน
- 3.1.3 ศึกษาวิธีการทดสอบซอฟต์แวร์ วิธีการสร้างกรณีทดสอบ และมาตรฐานสำหรับ
กรณีทดสอบ
- 3.1.4 ออกแบบฐานข้อมูล สำหรับจัดเก็บกรณีทดสอบ
- 3.1.5 พัฒนาส่วนรับข้อกำหนดความต้องการซอฟต์แวร์ และส่วนรับข้อมูลกรณีทดสอบ
- 3.1.6 รวบรวมฐานข้อมูลกรณีทดสอบ และสร้างกรณีทดสอบ

3.1.7 ทดสอบผลการรวม และแก้ไขข้อผิดพลาด โดยเครื่องคอมพิวเตอร์ที่ใช้ในการทดสอบเป็นคอมพิวเตอร์ Laptop CPU Pentium IV ความเร็ว 1.4 GHz หน่วยความจำหลัก 768 MB ฮาร์ดดิสก์ความจุ 40 GB

3.2 การวิเคราะห์และออกแบบเครื่องมือช่วยในการสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ

ในกระบวนการออกแบบและสร้างกรณีทดสอบ ผู้ทดสอบมีหน้าที่ในการศึกษาและออกแบบกรณีทดสอบ เพื่อให้ตรงตามวัตถุประสงค์ของการทดสอบมากที่สุดซึ่งขั้นตอนในการออกแบบและสร้างกรณีทดสอบเป็นขั้นตอนที่ย่างยาก ใช้เวลานาน และกรณีทดสอบที่สร้างขึ้นยังมีเป็นจำนวนมากทำให้ยากในการตรวจสอบ ซึ่งเครื่องมือที่พัฒนาขึ้นสามารถช่วยผู้ทดสอบจัดการกับปัญหาข้างต้นได้ ซึ่งเป็นการช่วยลดภาระของผู้ทดสอบ

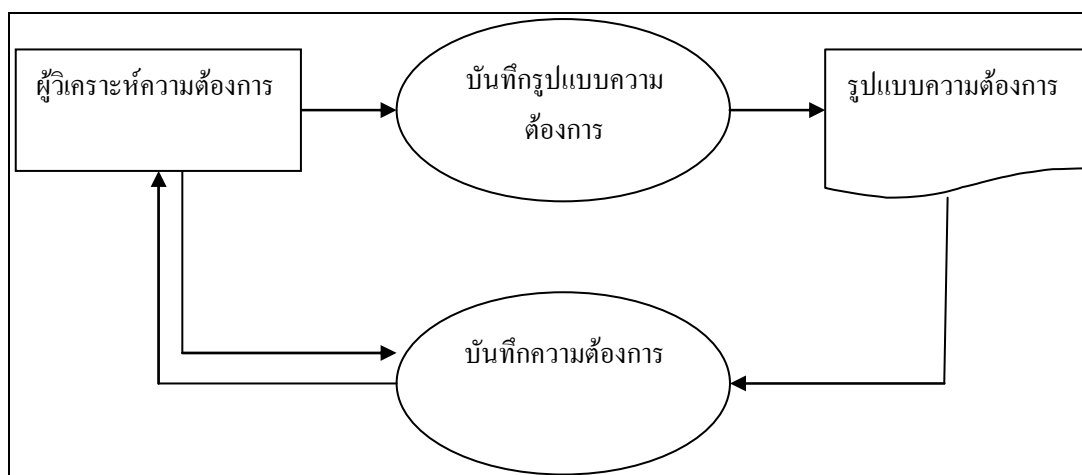
เครื่องมือที่พัฒนามีองค์ประกอบหลักอยู่ 3 ส่วนคือ ส่วนรับข้อกำหนดความต้องการซอฟต์แวร์ ส่วนรับข้อมูลกรณีทดสอบ และฐานข้อมูลกรณีทดสอบ ดังแสดงในรูปที่ 3.1



รูปที่ 3.1 องค์ประกอบหลักของเครื่องมือ

3.2.1 ส่วนรับข้อกำหนดความต้องการซอฟต์แวร์

ในส่วนนี้จะเป็นการเก็บรวบรวมข้อกำหนดความต้องการซอฟต์แวร์ของโครงการแต่ละโครงการ โดยมีขั้นตอนดังรูปที่ 3.2 และสามารถอธิบายถึงขั้นตอนการทำงานได้ดังนี้คือ

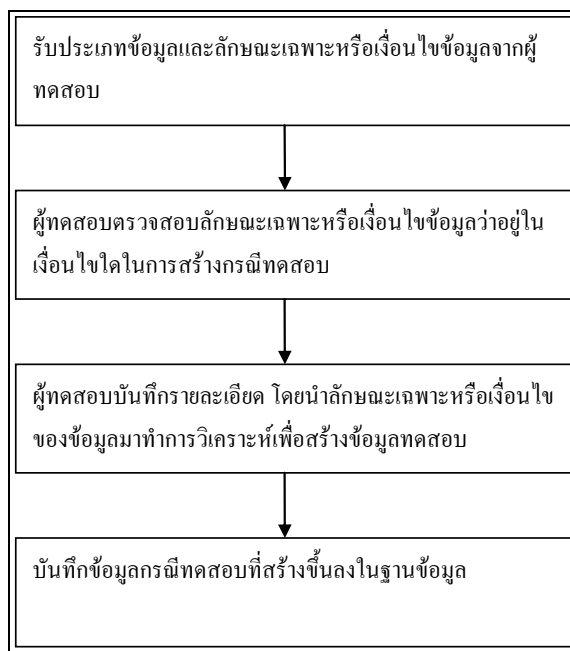


รูปที่ 3.2 แผนผังการทำงานของส่วนรับข้อกำหนดความต้องการซอฟต์แวร์

ผู้วิเคราะห์ความต้องการต้องรวบรวมความต้องการซอฟต์แวร์จากลูกค้าและทำการเพิ่มข้อกำหนดความต้องการ โดยใส่ข้อมูลสำหรับข้อกำหนดความต้องการตามรูปแบบของข้อกำหนดความต้องการที่ได้กำหนดไว้

3.2.2 ส่วนรับข้อมูลกรณีทดสอบ

ในส่วนนี้จะรับข้อมูลที่ถูกต้องและไม่ถูกต้องจากผู้ทดสอบ เพื่อนำมาสร้างกรณีทดสอบตามเงื่อนไขที่ได้รับ และส่วนของผลที่คาดว่าจะได้รับการทดสอบจะกำหนดได้เพียงว่าข้อมูลเข้าเป็นข้อมูลที่ถูกต้องหรือไม่ถูกต้อง จากนั้นระบบจะบันทึกกรณีทดสอบที่สร้างลงในฐานข้อมูลกรณีทดสอบ ผังรายละเอียดลำดับขั้นตอนการสร้างกรณีทดสอบแสดงในรูปที่ 3.3



รูปที่ 3.3 แสดงลำดับขั้นตอนการสร้างกรณีทดสอบ

3.2.3 ฐานข้อมูลกรณีทดสอบ

เป็นส่วนที่เก็บข้อมูลกรณีทดสอบ และข้อมูลอื่น ๆ ที่เกี่ยวกับกรณีทดสอบ โดยข้อมูลที่ผู้ทดสอบทำการบันทึกจะมีรูปแบบไมโครซอฟต์เวิร์ด หน้าที่หลักของฐานข้อมูลกรณีทดสอบมีดังนี้คือ

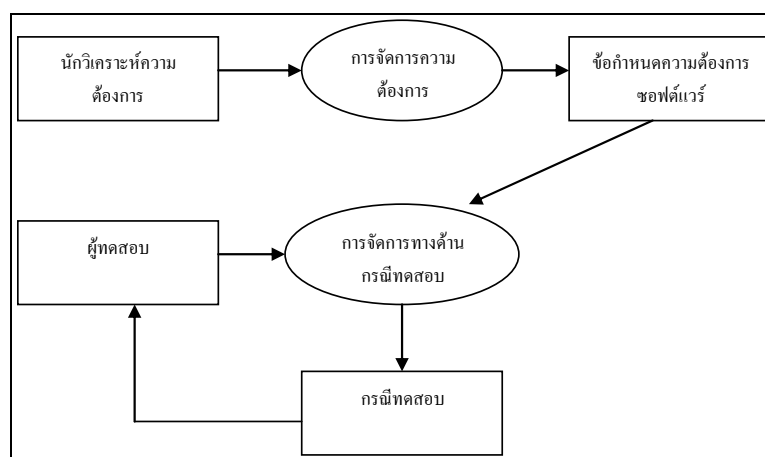
- ทำหน้าที่จัดการข้อมูลโครงการที่ทดสอบโดยผู้ทดสอบสามารถเพิ่ม ลด และแก้ไข ข้อมูลที่บันทึกอยู่ในระบบได้
- ทำหน้าที่จัดการข้อมูลกรณีทดสอบ และผลการทดสอบโดยผู้ทดสอบสามารถเพิ่ม ลด และแก้ไขกรณีทดสอบและผลการทดสอบที่บันทึกอยู่ในระบบได้
- ทำหน้าที่จัดการข้อมูลผู้ทดสอบโดยผู้ทดสอบสามารถ เพิ่ม ลด และแก้ไขข้อมูลผู้ทดสอบที่บันทึกอยู่ในระบบได้
- ประมวลผลการทดสอบซอฟต์แวร์ เป็นส่วนของการนำข้อมูลผลการทดสอบซอฟต์แวร์ที่ผู้ทดสอบบันทึกเข้าสู่ระบบ มาทำการประมวลผล เพื่อสรุปความก้าวหน้าของการทดสอบและ ทำการออกรายงานให้กับผู้ทดสอบ

- ประมวลผลการทำงานของผู้ทดสอบแต่ละคน โดยนำข้อมูลการทดสอบของผู้ทดสอบแต่ละคนมาสรุปผลและออกเอกสารรายงานการทำงานของผู้ทดสอบแต่ละคน
- เก็บข้อมูลทดสอบที่นำไปใช้สร้างกรณีทดสอบสำหรับชั้นสมมูลบางประเภท

3.2.4 การวิเคราะห์และออกแบบเครื่องมือ

ในการทดสอบแต่ละโครงการ ผู้ทดสอบเป็นผู้บันทึกข้อมูลของการทดสอบเข้าสู่ระบบ จากนั้นในขั้นตอนการออกแบบกรณีทดสอบผู้ทดสอบต้องศึกษารายละเอียดและวัตถุประสงค์ของการทดสอบ เพื่อให้กรณีทดสอบที่สร้างขึ้นครอบคลุมข้อกำหนดความต้องการที่ระบุไว้ รายละเอียดที่ผู้ทดสอบต้องนำไปใช้ออกแบบกรณีทดสอบ คือ ประเภทข้อมูล และลักษณะเฉพาะหรือเงื่อนไขของข้อมูลที่จะสร้างกรณีทดสอบ ตัวอย่าง เช่น ข้อมูลเข้าประเภทจำนวนจริง ซึ่งมีค่า มากกว่าศูนย์ ในการทดสอบแบบเบิ้ลบ็อกซ์ ข้อมูลส่วนนี้จะนำไปใช้ในการสร้างชั้นสมมูล ข้อมูลทดสอบ และ กรณีทดสอบตามลำดับ เครื่องมือจะรับข้อมูลเหล่านี้จากผู้ทดสอบ เพื่อนำไปสร้างกรณีทดสอบ

จากการวิเคราะห์และออกแบบเครื่องมือ ความสัมพันธ์ระหว่างองค์ประกอบต่าง ๆ ของเครื่องมือ กับข้อมูลที่ใช้ภายในระบบ สามารถสร้างเป็นแผนภาพกระแสข้อมูล ได้ดังรูปที่ 3.4



รูปที่ 3.4 แผนภาพกระแสข้อมูลของเครื่องมือ

จากแผนภาพกระแสข้อมูลของเครื่องมือ มีกระบวนการที่สำคัญคือ การจัดการความต้องการ และการจัดการกรณีทดสอบ

การจัดการความต้องการ เป็นส่วนที่รับข้อมูลรายละเอียดความต้องการจากนักวิเคราะห์ความต้องการ จากนั้นจึงบันทึกข้อมูลลงในฐานข้อมูล เพื่อให้ผู้ทดสอบใช้ประกอบในการสร้างกรณีทดสอบ

การจัดการกรณีทดสอบ ทำหน้าที่รับประเภทข้อมูล ข้อกำหนดและเงื่อนไขของข้อมูล เพื่อใช้สร้างกรณีทดสอบ และบันทึกกรณีทดสอบที่สร้างขึ้นลงในฐานข้อมูล

3.3 การออกแบบและสร้างฐานข้อมูล สำหรับเครื่องมือช่วยในการสร้างกรณีทดสอบ สำหรับการทดสอบการยอมรับ

ส่วนนี้แสดงถึงโครงสร้างฐานข้อมูลที่ใช้ในงานวิจัยนี้ ซึ่งเป็นระบบฐานข้อมูลเชิงสัมพันธ์ (Relational database) โดยใช้โปรแกรมออร์เคิลในการจัดเก็บข้อมูล ประกอบไปด้วยตาราง 58 ตาราง ซึ่งในแต่ละตารางประกอบด้วย ชื่อเขตข้อมูล (Field name) ชนิดของข้อมูล (Data type) ขนาดของข้อมูล (Field size) และหมายเหตุ พร้อมทั้งระบุถึงคีย์หลัก (Primary Key)

ตาราง Emp

ตาราง Emp ใช้เก็บรายละเอียดของพนักงาน โดยมีคีย์หลัก คือ รหัสพนักงาน (Empno) รายละเอียดดังตารางที่ 3.1

ตารางที่ 3.1 รายละเอียดในตาราง Emp

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Empno (PK)	ตัวอักษร	5	รหัสพนักงาน
Empname	ตัวอักษร	60	ชื่อพนักงาน
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Customer

ตาราง Customer ใช้เก็บรายละเอียดของลูกค้า โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รายละเอียดดังตารางที่ 3.2

ตารางที่ 3.2 รายละเอียดในตาราง Customer

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Cust_name	ตัวอักษร	60	ชื่อลูกค้า
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Cust_release

ตาราง Cust_release เก็บรายละเอียดของ release โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) และ รหัสrelease (Release_code) รายละเอียดดังตารางที่ 3.3

ตารางที่ 3.3 รายละเอียดในตาราง Cust_release

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Release_name	ตัวอักษร	60	ชื่อrelease
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Project

ตาราง Project ใช้เก็บรายละเอียดของโครงการ โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) และรหัสโครงการ (Project_code) รายละเอียดดังตารางที่ 3.4

ตารางที่ 3.4 รายละเอียดในตาราง Project

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
Project_name	ตัวอักษร	60	ชื่อโครงการ
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Project_head

ตาราง Project_head ใช้เก็บรายละเอียดของข้อกำหนดความต้องการซอฟต์แวร์ โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) และรหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) รายละเอียดดังตารางที่ 3.5

ตารางที่ 3.5 รายละเอียดในตาราง Project_head

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
SRS_description	ตัวอักษร	200	รายละเอียดข้อกำหนดความต้องการ
Ref_no	ตัวอักษร	20	เอกสารอ้างอิง
Prepare_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Prepare_date	วันที่	-	วันที่ปรับปรุงข้อมูล
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec01_Abbreviations

ตาราง Sec01_Abbreviations ใช้เก็บรายละเอียดของคำย่อของคำศัพท์ต่าง ๆ ที่ใช้ใน SRS โดยไม่มีกฎเกณฑ์ในการย่อ เพื่อสะดวกในการใช้งาน (เช่น Std ย่อมาจาก Standard) อาจจะนำคำย่อนี้ไว้ในส่วนของภาคผนวก หรือใช้อ้างอิงไปยังเอกสารอื่น ๆ ได้ โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสคำย่อ (Abbreviation_code) รายละเอียดดังตารางที่ 3.6

ตารางที่ 3.6 รายละเอียดในตาราง Sec01_Abbreviations

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Abbreviation_code (PK)	ตัวอักษร	20	รหัสคำย่อ
Abbreviation_description	ตัวอักษร	200	รายละเอียดคำย่อ
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec01_Acronyms

ตาราง Sec01_Acronyms ใช้เก็บรายละเอียดคำย่อของคำศัพท์ต่าง ๆ ที่ใช้ใน SRS โดยนำเอาอักษรตัวแรกของคำศัพท์มาย่อ เพื่อให้สะดวกในการใช้งาน (เช่น SRS ย่อมาจาก Software Requirements Specification) อาจจะนำคำย่อนี้ไว้ในส่วนของภาคผนวก หรือใช้อ้างอิงไปยังเอกสารอื่น ๆ ได้ โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสคำย่อ (Acronyms_code) รายละเอียดดังตารางที่ 3.7

ตารางที่ 3.7 รายละเอียดในตาราง Sec01_Acronyms

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Acronyms_code (PK)	ตัวอักษร	20	รหัสคำย่อ
Acronyms_description	ตัวอักษร	200	รายละเอียดคำย่อ
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec01_Definition

ตาราง Sec01_Definition ใช้เก็บรายละเอียดของคำจำกัดความของคำศัพท์ต่าง ๆ ที่ใช้ใน SRS เพื่อให้เข้าใจความหมายของคำศัพท์ที่ตรงกัน โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัส release (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสคำจำกัดความ (Definition_code) รายละเอียดดังตารางที่ 3.8

ตารางที่ 3.8 รายละเอียดในตาราง Sec01_Definition

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Definition_code (PK)	ตัวอักษร	20	รหัสคำจำกัดความ
Definition_description	ตัวอักษร	200	รายละเอียดคำจำกัดความ
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec01_Overview

ตาราง Sec01_Overview ใช้เก็บรายละเอียดเนื้อหาโดยรวมที่อยู่ใน SRS เป็นการมองภาพรวมของเอกสาร และอธิบายว่า SRS มีการจัดระเบียบเอกสารอย่างไร โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสเนื้อหาโดยรวม (Overview_code) รายละเอียดดังตารางที่ 3.9

ตารางที่ 3.9 รายละเอียดในตาราง Sec01_Overview

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Overview_code (PK)	ตัวอักษร	20	รหัสเนื้อหาโดยรวม
Overview_description	ตัวอักษร	200	รายละเอียดเนื้อหาโดยรวม
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec01_Purpose

ตาราง Sec01_Purpose ใช้เก็บรายละเอียดของจุดประสงค์ของการเขียน SRS หรือความตั้งใจที่เฉพาะเจาะจงของผู้ที่จะใช้ SRS โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสจุดประสงค์ (Purpose_code) รายละเอียดดังตารางที่ 3.10

ตารางที่ 3.10 รายละเอียดในตาราง Sec01_Purpose

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Purpose_code (PK)	ตัวอักษร	20	รหัสจุดประสงค์
Purpose_description	ตัวอักษร	200	รายละเอียดจุดประสงค์
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec01_References

ตาราง Sec01_References ใช้เก็บรายละเอียดของเอกสารอ้างอิงที่ใช้ใน SRS โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสเอกสารอ้างอิง (References_code) รายละเอียดดังตารางที่ 3.11

ตารางที่ 3.11 รายละเอียดในตาราง Sec01_References

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Reference_code (PK)	ตัวอักษร	20	รหัสเอกสารอ้างอิง
Reference_description	ตัวอักษร	200	รายละเอียดเอกสารอ้างอิง
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec01_Scope

ตาราง Sec01_Scope ใช้เก็บรายละเอียดขอบเขตของซอฟต์แวร์ที่กำหนดไว้ ซึ่งจะต้องประกอบด้วย ชื่อของซอฟต์แวร์ (เช่น ระบบ ABC) หน้าที่ของซอฟต์แวร์ SRS จะต้องมีความสอดคล้องกับเอกสารระดับสูง (เช่น System Requirements Specification : SyRS) และอธิบายถึงการนำเอาซอฟต์แวร์ไปประยุกต์ใช้ให้บรรลุตามวัตถุประสงค์ที่กำหนด ประโยชน์ที่คาดว่าจะได้รับ และเป้าหมายที่ตั้งไว้ โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสขอบเขต (Scope_code) รายละเอียดดังตารางที่ 3.12

ตารางที่ 3.12 รายละเอียดในตาราง Sec01_Scope

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Scope_code (PK)	ตัวอักษร	20	รหัสขอบเขต
Scope_description	ตัวอักษร	200	รายละเอียดขอบเขต
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec02_Apport_Req

ตาราง Sec02_Apport_Req ใช้เก็บรายละเอียดของความต้องการที่ผู้ใช้หรือลูกค้ายอมให้เกิดความล่าช้า หรือเป็นความต้องการที่คาดว่าจะไม่สามารถพัฒนาเสร็จในเวอร์ชันนี้ และจะทำการพัฒนาซอฟต์แวร์ในเวอร์ชันต่อไปในอนาคต เพื่อให้ทำหน้าที่ครอบคลุมตามความต้องการของผู้ใช้หรือลูกค้าได้อย่างครบถ้วน โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสการแบ่งสัณปันส่วนความต้องการ (Apport_Req_code) รายละเอียดดังตารางที่ 3.13

ตารางที่ 3.13 รายละเอียดในตาราง Sec02_Apport_Req

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Apport_Req_code (PK)	ตัวอักษร	20	รหัสการแบ่งสัณป็นส่วนความต้องการ
Apport_Req_description	ตัวอักษร	200	รายละเอียดการแบ่งสัณป็นส่วนความต้องการ
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec02_Assump_Depend

ตาราง Sec02_Assump_Depend ใช้เก็บรายละเอียดของปัจจัยต่าง ๆ ที่ส่งผลกระทบต่อความต้องการที่ระบุไว้ใน SRS เช่น ตั้งข้อสมมุติฐานว่าจะใช้ระบบการจัดการฐานข้อมูล ORACLE 10G กับซอฟต์แวร์นี้ แต่ในความเป็นจริง เนื่องจาก ORACLE 10G ยังพัฒนาไม่เสร็จ ทำให้ซอฟต์แวร์ที่พัฒนาขึ้นมาไม่สามารถใช้กับระบบจัดการฐานข้อมูล ORACLE 10G ได้ เป็นต้น ดังนั้นความต้องการที่ระบุไว้ใน SRS ต้องได้รับการเปลี่ยนแปลงด้วย โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสข้อสันนิษฐานและการขึ้นต่อกัน (Assump_Depend_code) รายละเอียดดังตารางที่ 3.14

ตารางที่ 3.14 รายละเอียดในตาราง Sec02_Assump_Depend

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Assump_Depend_code (PK)	ตัวอักษร	20	รหัสข้อสันนิษฐานและการขึ้นต่อกัน
Assump_Depend_description	ตัวอักษร	200	รายละเอียดข้อสันนิษฐานและการขึ้นต่อกัน
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec02_Comm_interface

ตาราง Sec02_Comm_Interface ใช้เก็บรายละเอียดการต่อประสานกับระบบสื่อสารอื่น ๆ ในรูปแบบต่าง ๆ เช่น local network protocols เป็นต้น โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสการต่อประสานกับระบบสื่อสาร (Comm_code) รายละเอียดดังตารางที่ 3.15

ตารางที่ 3.15 รายละเอียดในตาราง Sec02_Comm_Interface

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Comm_code (PK)	ตัวอักษร	20	รหัสการต่อประสานกับระบบสื่อสาร
Comm_description	ตัวอักษร	200	รายละเอียดการต่อประสานกับระบบสื่อสาร
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec02_Constraint

ตาราง Sec02_Constraint ใช้เก็บรายละเอียดของคำอธิบายและข้อจำกัดในการพัฒนาซอฟต์แวร์ที่ควรคำนึงถึง คือ นโยบายทั่ว ๆ ไป ข้อจำกัดทางฮาร์ดแวร์ การต่อประสานไปยังระบบอื่น การปฏิบัติการแบบขนาน หน้าที่การตรวจสอบระบบ หน้าที่การควบคุมระบบ ความต้องการในเรื่อง higher-order language signal handshake protocols (เช่น XON-XOFF, ACK-NACK) ความต้องการความเชื่อถือได้ ความวิกฤตในการประยุกต์ใช้ระบบ (เช่น มี transaction เข้ามาในระบบมาก ๆ) และพิจารณาถึงความปลอดภัยและความมั่นคง โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสข้อจำกัด (Constraint_code) รายละเอียดดังตารางที่ 3.16

ตารางที่ 3.16 รายละเอียดในตาราง Sec02_Constraint

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Constraint_code (PK)	ตัวอักษร	20	รหัสข้อจำกัด
Constraint_description	ตัวอักษร	200	รายละเอียดข้อจำกัด
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec02_Hard_Interface

ตาราง Sec02_Hard_Interface ใช้เก็บรายละเอียดของการต่อประสานระหว่างซอฟต์แวร์กับส่วนประกอบของฮาร์ดแวร์ เช่น number of port, instruction set, devices และ protocols เป็นต้น โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสการต่อประสานกับฮาร์ดแวร์ (Hard_code) รายละเอียดดังตารางที่ 3.17

ตารางที่ 3.17 รายละเอียดในตาราง Sec02_Hard_Interface

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Hard_code (PK)	ตัวอักษร	20	รหัสการต่อประสานกับฮาร์ดแวร์
Hard_description	ตัวอักษร	200	รายละเอียดต่อประสานกับฮาร์ดแวร์
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec02_Mem_Constraint

ตาราง Sec02_Mem_Constraint ใช้เก็บรายละเอียดลักษณะและข้อจำกัดของหน่วยความจำหลักและหน่วยความจำรอง โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสข้อจำกัดของหน่วยความจำ (Mem_code) รายละเอียดดังตารางที่ 3.18

ตารางที่ 3.18 รายละเอียดในตาราง Sec02_Mem_Constraint

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Mem_code (PK)	ตัวอักษร	20	รหัสข้อจำกัดของหน่วยความจำ
Mem_description	ตัวอักษร	200	รายละเอียดข้อจำกัดของหน่วยความจำ
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล

ตาราง Sec02_Operation

ตาราง Sec02_Operation ใช้เก็บรายละเอียดของการดำเนินการใช้งานซอฟต์แวร์แบบปกติ และแบบกรณีพิเศษ เช่น การดำเนินการของผู้ใช้เพื่อเริ่มต้นใช้ระบบ ระยะเวลาของการดำเนินการแบบ interactive ระยะเวลาของการดำเนินการที่ไม่มีการสนใจเครื่องหรือระบบ ฟังก์ชันสนับสนุนการประมวลผลข้อมูล (เช่น การเรียงลำดับข้อมูล) การดำเนินการสำรองและกู้คืนข้อมูล เป็นต้น โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสการดำเนินการ (Operation_code) รายละเอียดดังตารางที่ 3.19

ตารางที่ 3.19 รายละเอียดในตาราง Sec02_Operation

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Operation_code (PK)	ตัวอักษร	20	รหัสการดำเนินการ
Operation_description	ตัวอักษร	200	รายละเอียดการดำเนินการ
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec02_Prod_Func

ตาราง Sec02_Prod_Func ใช้เก็บรายละเอียดของหน้าที่หลักของซอฟต์แวร์ เพื่อสร้างความเข้าใจให้กับผู้ใช้และคนอื่นที่อ่านเอกสารนี้เป็นครั้งแรก โดยอาจใช้ข้อความหรือรูปภาพก็ได้ โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสหน้าที่ของซอฟต์แวร์ (Prod_Func_code) รายละเอียดดังตารางที่ 3.20

ตารางที่ 3.20 รายละเอียดในตาราง Sec02_Prod_Func

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Prod_Func_code (PK)	ตัวอักษร	20	รหัสหน้าที่ของซอฟต์แวร์
Prod_Func_description	ตัวอักษร	200	รายละเอียดหน้าที่ของซอฟต์แวร์
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec02_Qua_Prov

ตาราง Sec02_Qua_Prov ใช้เก็บรายละเอียดของกระบวนการที่จะนำมาพิจารณาซอฟต์แวร์ว่าซอฟต์แวร์มีความเหมาะสมในการใช้งานหรือไม่ เช่น ความสะดวก ค่าใช้จ่าย และระยะเวลาของการใช้งาน เป็นต้น โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสข้อกำหนดในการเตรียมกระบวนการในการพิจารณาซอฟต์แวร์ (Qua_Prov_code) รายละเอียดดังตารางที่ 3.21

ตารางที่ 3.21 รายละเอียดในตาราง Sec02_Qua_Prov

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Qua_Prov_code (PK)	ตัวอักษร	20	รหัสข้อจำกัดในการเตรียม กระบวนการในการพิจารณา ซอฟต์แวร์
Qua_Prov_description	ตัวอักษร	200	รายละเอียดข้อจำกัดในการเตรียม กระบวนการในการพิจารณา ซอฟต์แวร์
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec02_Req_Trace

ตาราง Sec02_Req_Trace ใช้เก็บรายละเอียดของความสามารถในการตรวจสอบความต้องการกลับไปได้ว่าความต้องการนั้น ๆ มีจุดกำเนิดมาจากใคร มีจุดประสงค์อะไร เพื่อใช้อ้างอิงต่อไปในอนาคต เช่น เมื่อมีการเปลี่ยนแปลงความต้องการ ทำให้ผู้พัฒนาซอฟต์แวร์ต้องทำการปรับปรุงซอฟต์แวร์ ซึ่งผู้พัฒนาซอฟต์แวร์สามารถตรวจสอบความต้องการนั้น ๆ กลับไปได้ว่าใครเป็นผู้พัฒนามาก่อนหน้านี้ เพื่อสอบถามข้อมูลต่าง ๆ จากผู้นั้นได้ โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสความสามารถในการติดตามความต้องการ (Req_Trace_code) รายละเอียดดังตารางที่ 3.22

ตารางที่ 3.22 รายละเอียดในตาราง Sec02_Req_Trace

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Req_Trace_code (PK)	ตัวอักษร	20	รหัสความสามารถในการติดตามความต้องการ
Req_Trace_description	ตัวอักษร	200	รายละเอียดความสามารถในการติดตามความต้องการ
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec02_Site_Adap_Req

ตาราง Sec02_Site_Adap_Req ใช้เก็บรายละเอียดที่ตั้งของการติดตั้งซอฟต์แวร์ รวมทั้งวิธีการดำเนินการที่จะใช้กับซอฟต์แวร์ เช่น พิจารณาถึง config เพื่อให้ซอฟต์แวร์เข้ากับเครื่องคอมพิวเตอร์นั้น ๆ ได้ พิจารณาถึง platform จำนวน node และ port ที่ใช้กับซอฟต์แวร์นี้ เป็นต้น ถ้าที่ตั้งอยู่ในสภาพแวดล้อมที่ไม่เหมาะสม อาจเปลี่ยนแปลงสภาพแวดล้อมให้เหมาะสมได้ โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสความต้องการด้านการติดตั้งซอฟต์แวร์ (Site_Adap_Req_code) รายละเอียดดังตารางที่ 3.23

ตารางที่ 3.23 รายละเอียดในตาราง Sec02_Site_Adap_Req

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Site_Adap_Req_code (PK)	ตัวอักษร	20	รหัสความต้องการด้านการติดตั้งซอฟต์แวร์
Site_Adap_Req_description	ตัวอักษร	200	รายละเอียดความต้องการด้านการติดตั้งซอฟต์แวร์
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec02_Soft_Interface

ตาราง Sec02_Soft_Interface ใช้เก็บรายละเอียดการต่อประสานระหว่างซอฟต์แวร์กับระบบอื่น เช่น การเชื่อมโยงระหว่างระบบลูกหนึ่กับระบบบัญชีแยกประเภท โดยซอฟต์แวร์แต่ละชนิด ประกอบด้วย ชื่อย่อ (mnemonic) หมายเลขรุ่น (version) และแหล่งที่มา โดยมีคีย์หลัก คือรหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสการต่อประสานกับซอฟต์แวร์ (Soft_code) รายละเอียดดังตารางที่ 3.24

ตารางที่ 3.24 รายละเอียดในตาราง Sec02_Soft_Interface

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Soft_code (PK)	ตัวอักษร	20	รหัสการต่อประสานกับซอฟต์แวร์
Soft_description	ตัวอักษร	200	รายละเอียดการต่อประสานกับซอฟต์แวร์
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec02_User_Interface

ตาราง Sec02_User_Interface ใช้เก็บรายละเอียดของการต่อประสานระหว่างซอฟต์แวร์กับผู้ใช้หรือลูกค้า โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสการต่อประสานกับผู้ใช้ (User_Interface_code) รายละเอียดดังตารางที่ 3.25

ตารางที่ 3.25 รายละเอียดในตาราง Sec02_User_Interface

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
User_Interface_code (PK)	ตัวอักษร	20	รหัสการต่อประสานกับผู้ใช้
User_Interface_description	ตัวอักษร	200	รายละเอียดการต่อประสานกับผู้ใช้
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec03_Acceptance_Req

ตาราง Sec03_Acceptance_Req ใช้เก็บรายละเอียดของข้อกำหนดในการตรวจรับซอฟต์แวร์เพื่อตรวจสอบว่าซอฟต์แวร์สามารถทำหน้าที่ได้ตรงตามความต้องการของลูกค้าหรือไม่ โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสความต้องการด้านการยอมรับ (Acceptance_Req_code) รายละเอียดดังตารางที่ 3.26

ตารางที่ 3.26 รายละเอียดในตาราง Sec03_Acceptance_Req

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Acceptance_Req_code (PK)	ตัวอักษร	20	รหัสความต้องการด้านการยอมรับ
Acceptance_Req_description	ตัวอักษร	200	รายละเอียดความต้องการด้านการยอมรับ
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec03_Availability

ตาราง Sec03_Availability ใช้เก็บรายละเอียดปัจจัยที่จำเป็นต้องมี เพื่อรองรับสภาพพร้อมใช้งานของซอฟต์แวร์โดยรวมในระดับที่กำหนดไว้ เช่น Checkpoint และ Recovery เป็นต้น โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสสภาพพร้อมใช้งาน (Availability_code) รายละเอียดดังตารางที่ 3.27

ตารางที่ 3.27 รายละเอียดในตาราง Sec03_Availability

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Availability_code (PK)	ตัวอักษร	20	รหัสสภาพพร้อมใช้งาน
Availability_description	ตัวอักษร	200	รายละเอียดสภาพพร้อมใช้งาน
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec03_Comp_Hard_Resource_Req

ตาราง Sec03_Comp_Hard_Resource_Req ใช้เก็บรายละเอียดของการใช้ทรัพยากรฮาร์ดแวร์คอมพิวเตอร์ภายในระบบ เพื่อให้เกิดประโยชน์สูงสุดในการใช้งาน เช่นความสามารถในการใช้โปรเซสเซอร์ หน่วยความจำ อุปกรณ์นำเข้าหรือออก อุปกรณ์หน่วยความจำรอง อุปกรณ์การสื่อสารข้อมูล อุปกรณ์เครือข่าย เป็นต้น โดยมีคีย์หลักคือรหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสความต้องการด้านการใช้ทรัพยากรฮาร์ดแวร์คอมพิวเตอร์ (Comp_Hard_Resource_code) รายละเอียดดังตารางที่ 3.28

ตารางที่ 3.28 รายละเอียดในตาราง Sec03_Comp_Hard_Resource_Req

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Comp_Hard_Resource_code (PK)	ตัวอักษร	20	รหัสความต้องการด้านการใช้ทรัพยากรฮาร์ดแวร์คอมพิวเตอร์
Comp_Hard_Resource_description	ตัวอักษร	200	รายละเอียดความต้องการด้านการใช้ทรัพยากรฮาร์ดแวร์คอมพิวเตอร์
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec03_Doc_Req

ตาราง Sec03_Doc_Req ใช้เก็บรายละเอียดของเอกสารที่ใช้ในการพัฒนาซอฟต์แวร์ ซึ่งอยู่นอกเหนือจากเอกสารที่กำหนดไว้แล้วในมาตรฐาน เช่น รูปแบบโดยละเอียดของเอกสารคู่มือผู้ใช้ซอฟต์แวร์ (Software User Manual) เป็นต้น โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสความต้องการด้านเอกสาร (Doc_Req_code) รายละเอียดดังตารางที่ 3.29

ตารางที่ 3.29 รายละเอียดในตาราง Sec03_Doc_Req

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Doc_Req_code (PK)	ตัวอักษร	20	รหัสความต้องการด้านเอกสาร
Doc_Req_description	ตัวอักษร	200	รายละเอียดความต้องการด้านเอกสาร
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec03_Ext_Interface

ตาราง Sec03_Ext_Interface ใช้เก็บรายละเอียดของรายละเอียดของข้อมูลที่เข้าไปยังระบบ และข้อมูลที่ออกจากระบบทั้งหมด ประกอบด้วย ชื่อของตัวข้อมูล (item) จุดประสงค์ แหล่งที่มา ของข้อมูล ช่วงของข้อมูลที่ต้อง (valid) หน่วยของข้อมูล (เช่น จำนวนเพิ่มข้อมูล จำนวน record จำนวนเอกสาร) ระยะเวลา ความสัมพันธ์ระหว่างข้อมูลเข้าและข้อมูลออก รูปแบบหน้าจอ รูปแบบ วินโดวส์ รูปแบบของข้อมูล รูปแบบของคำสั่งงาน ข้อความจบ (end message) โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสการต่อประสานภายนอก (Ext_Interf_code) รายละเอียดดังตารางที่ 3.30

ตารางที่ 3.30 รายละเอียดในตาราง Sec03_Ext_Interface

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Ext_Interf_code (PK)	ตัวอักษร	20	รหัสการต่อประสานภายนอก
Ext_Interf_description	ตัวอักษร	200	รายละเอียดการต่อประสานภายนอก
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec03_Functional_Requirement

ตาราง Sec03_Functional_Requirement ใช้เก็บรายละเอียดของหน้าที่ของซอฟต์แวร์ว่าซอฟต์แวร์ต้องทำงานอย่างไร เมื่อมีการรับข้อมูลเข้า ทำการประมวลผลและสร้างข้อมูลออก โดยอธิบายในลักษณะของประโยคที่ว่า "ระบบควรจะ...." ประกอบด้วย การตรวจสอบความถูกต้องของข้อมูลเข้า ลำดับที่แน่นอนของการดำเนินการ การตอบสนองต่อเหตุการณ์ที่ผิดปกติ (เช่น การเกิด overflow การจัดการด้านการสื่อสาร การจัดการในส่วนที่เป็น error และสามารถแก้ไขได้) ความสัมพันธ์ระหว่างข้อมูลเข้าและข้อมูลออก (เช่น ลำดับของข้อมูลเข้าและข้อมูลออก การแปลงข้อมูลเข้าไปเป็นข้อมูลออก) โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสความต้องการด้านหน้าที่ (Func_Req_code) รายละเอียดดังตารางที่ 3.31

ตารางที่ 3.31 รายละเอียดในตาราง Sec03_Functional_Requirement

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Func_Req_code (PK)	ตัวอักษร	20	รหัสความต้องการด้านหน้าที่
Func_Req_description	ตัวอักษร	200	รายละเอียดความต้องการด้านหน้าที่
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec03_Implementation_Constrain

ตาราง Sec03_Implementation_Constrain ใช้เก็บรายละเอียดของข้อบังคับที่ใช้ในการออกแบบและการใช้ซอฟต์แวร์และฮาร์ดแวร์ร่วมกัน เพื่อสนับสนุนการใช้งานของซอฟต์แวร์ที่พัฒนา เช่น ใช้ตัวแปลภาษาโดยเฉพาะ หรือซอฟต์แวร์ต้อง run โดยใช้ระบบฐานข้อมูล หรือซอฟต์แวร์ต้องมีขนาดพอดีกับหน่วยความจำ 512 kbytes เป็นต้น โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสข้อจำกัดในการใช้งานซอฟต์แวร์ (Imp_Cons_code) รายละเอียดดังตารางที่ 3.32

ตารางที่ 3.32 รายละเอียดในตาราง Sec03_Implementation_Constrain

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Imp_Cons_code (PK)	ตัวอักษร	20	รหัสข้อจำกัดในการใช้งานซอฟต์แวร์
Imp_Cons_description	ตัวอักษร	200	รายละเอียดข้อจำกัดในการใช้งานซอฟต์แวร์
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec03_Logic_Dat_Req

ตาราง Sec03_Logic_Dat_Req ใช้เก็บรายละเอียดความต้องการเชิงตรรกะของข้อมูลที่จะทำการจัดเก็บในฐานข้อมูล โดยจะประกอบไปด้วย ชนิดข้อมูล ความถี่ของการใช้งาน วิธีในการเข้าถึงข้อมูล (เช่น SELECT, UPDATE, INSERT) ระบุ data entity และความสัมพันธ์ระหว่าง entity integrity constraints (เช่น เมื่อมีการ UPDATE ข้อมูล ต้องทำการ UPDATE ข้อมูลทุกตารางที่สัมพันธ์กัน) data retention requirements (เช่น กำหนดระยะเวลาในการเก็บข้อมูลไว้จนกว่าไม่ใช่ข้อมูลนั้น) โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสความต้องการด้านฐานข้อมูลเชิงตรรกะ (Logic_Dat_Req_code) รายละเอียดดังตารางที่ 3.33

ตารางที่ 3.33 รายละเอียดในตาราง Sec03_Logig_Dat_Req

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Logic_Dat_Req_code (PK)	ตัวอักษร	20	รหัสความต้องการด้านฐานข้อมูล เชิงตรรกะ
Logic_Dat_Req_description	ตัวอักษร	200	รายละเอียดความต้องการด้าน ฐานข้อมูลเชิงตรรกะ
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec03_Logistic_Related_Req

ตาราง Sec03_Logistic_Related_Req ใช้เก็บรายละเอียดของขั้นตอนต่าง ๆ ที่เกี่ยวข้องใน การนำซอฟต์แวร์มาใช้งาน (implementation) การสนับสนุนการใช้ซอฟต์แวร์ (support) และการ บำรุงรักษาซอฟต์แวร์ (maintenance) รวมถึงการจัดหาทรัพยากร เพื่อดำเนินการตามขั้นตอน ดังกล่าวด้วย โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสความต้องการด้าน การเตรียมขั้นตอนต่าง ๆ (Log_Related_Req_code) รายละเอียดดังตารางที่ 3.34

ตารางที่ 3.34 รายละเอียดในตาราง Sec03_Logistic_Related_Req

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Log_Related_Req_code (PK)	ตัวอักษร	20	รหัสความต้องการด้านการเตรียมขั้นตอนต่าง ๆ
Log_Related_Req_description	ตัวอักษร	200	รายละเอียดความต้องการด้านการเตรียมขั้นตอนต่าง ๆ
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec03_Maintainability

ตาราง Sec03_Maintainability ใช้เก็บรายละเอียดระดับความยากง่ายในการแก้ไขความผิดพลาด (fault) และการปรับเปลี่ยนซอฟต์แวร์ตามความต้องการของผู้ใช้งานหรือลูกค้า โดยระบุเป็นตัววัดเชิงปริมาณ ที่เรียกว่า mean time to repair a fault (MTTR) โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสความสามารถในการบำรุงรักษา (Maintainability_code) รายละเอียดดังตารางที่ 3.35

ตารางที่ 3.35 รายละเอียดในตาราง Sec03_Maintainability

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Maintainability_code (PK)	ตัวอักษร	20	รหัสความสามารถในการบำรุงรักษา
Maintainability_description	ตัวอักษร	200	รายละเอียดความสามารถในการบำรุงรักษา
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec03_Operate_Req

ตาราง Sec03_Operate_Req ใช้เก็บรายละเอียดของวิธีการดำเนินงานของซอฟต์แวร์และวิธีการติดต่อระหว่างผู้ใช้กับระบบ เช่น โครงร่างของหน้าจอ ข้อความผิดพลาด ระบบช่วยเหลือ เป็นต้น ซึ่งเนื้อหาในส่วนนี้จะประกอบขึ้นเป็นประโยชน์ในการนิยาม semantics และ syntax ของคำสั่ง โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสความต้องการด้านการดำเนินการ (Operate_Req_code) รายละเอียดดังตารางที่ 3.36

ตารางที่ 3.36 รายละเอียดในตาราง Sec03_Operate_Req

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Operate_Req_code (PK)	ตัวอักษร	20	รหัสความต้องการด้านการดำเนินการ
Operate_Req_description	ตัวอักษร	200	รายละเอียดความต้องการด้านการดำเนินการ
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec03_Packaging_Req

ตาราง Sec03_Packaging_Req ใช้เก็บรายละเอียดสื่อที่ใช้ในการเก็บบันทึกซอฟต์แวร์ (เช่น ดิสก์ เทป ซีดีรอม) เอกสารประกอบการใช้งานของซอฟต์แวร์ และอาจรวมถึงตัวต่อประสานที่จำเป็นต้องใช้กับซอฟต์แวร์ (เช่น I/O drivers) โดยมีคีย์หลักคือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสความต้องการด้านการบรรจุภัณฑ์ (Package_code) รายละเอียดดังตารางที่ 3.37

ตารางที่ 3.37 รายละเอียดในตาราง Sec03_Packaging_Req

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Package_code (PK)	ตัวอักษร	20	รหัสความต้องการด้านการบรรจุภัณฑ์
Package_description	ตัวอักษร	200	รายละเอียดความต้องการด้านการบรรจุภัณฑ์
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec03_Perform_Req

ตาราง Sec03_Perform_Req ใช้เก็บรายละเอียดของค่าที่ใช้วัดประสิทธิภาพของซอฟต์แวร์ โดยใช้กับตัวแปรที่สามารถวัดค่าได้ เช่น จำนวนของผู้ใช้ ขนาดข้อมูลที่อยู่ในฐานข้อมูล เวลาที่ใช้ในการโต้ตอบระหว่างผู้ใช้กับระบบ (response time) ปริมาณงาน (throughput) เป็นต้น โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสความต้องการด้านประสิทธิภาพ (Perform_Req_code) รายละเอียดดังตารางที่ 3.38

ตารางที่ 3.38 รายละเอียดในตาราง Sec03_Perform_Req

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Perform_Req_code (PK)	ตัวอักษร	20	รหัสความต้องการด้านประสิทธิภาพ
Perform_Req_description	ตัวอักษร	200	รายละเอียดความต้องการด้านประสิทธิภาพ
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec03_Portability

ตาราง Sec03_Portability ใช้เก็บรายละเอียดของระดับความยากง่ายในการย้ายซอฟต์แวร์เพื่อไปใช้งานบน platform อื่น เช่น เปอร์เซ็นต์ของการใช้ compiler และระบบปฏิบัติการ โดยเฉพาะ เพื่อสามารถใช้ซอฟต์แวร์กับเครื่องคอมพิวเตอร์อื่น ๆ ได้ เป็นต้น โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสความสามารถในการใช้ได้หลายระบบ (Portability_code) รายละเอียดดังตารางที่ 3.39

ตารางที่ 3.39 รายละเอียดในตาราง Sec03_Portability

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Portability_code (PK)	ตัวอักษร	20	รหัสความสามารถในการใช้ได้หลายระบบ
Portability_description	ตัวอักษร	200	รายละเอียดความสามารถในการใช้ได้หลายระบบ
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec03_Reliability

ตาราง Sec03_Reliability ใช้เก็บรายละเอียดเวลาที่เฉลี่ยที่ซอฟต์แวร์สามารถทำงานได้ก่อนมีความผิดพลาดเกิดขึ้น เรียกว่า mean time before failure (MTBF) เพื่อสร้างความเชื่อถือได้ในการส่งมอบซอฟต์แวร์ให้กับลูกค้า โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสความเชื่อถือได้ (Reliability_code) รายละเอียดดังตารางที่ 3.40

ตารางที่ 3.40 รายละเอียดในตาราง Sec03_Reliability

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Reliability_code (PK)	ตัวอักษร	20	รหัสความเชื่อถือได้
Reliability_description	ตัวอักษร	200	รายละเอียดความเชื่อถือได้
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec03_Safety_Req

ตาราง Sec03_Safety_Req ใช้เก็บรายละเอียดความปลอดภัยในระบบ เช่น ระบุถึงฟังก์ชันที่วิกฤต เพื่อลดโอกาสที่จะทำให้เกิดความเสียหายจากความผิดพลาดของซอฟต์แวร์ ซึ่งอาจเป็นอันตรายต่อคนหรือทรัพย์สิน โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสความต้องการด้านความปลอดภัย (Safety_Req_code) รายละเอียดดังตารางที่ 3.41

ตารางที่ 3.41 รายละเอียดในตาราง Sec03_Safety_Req

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Safety_Req_code (PK)	ตัวอักษร	20	รหัสความต้องการด้านความปลอดภัย
Safety_Req_description	ตัวอักษร	200	รายละเอียดความต้องการด้านความปลอดภัย
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec03_Security

ตาราง Sec03_Security ใช้เก็บรายละเอียดวิธี หรือเทคนิคต่าง ๆ เพื่อป้องกันเหตุที่เกิดจากความมั่วร้ายในการเข้าถึงข้อมูล เช่น กระบวนการในการเข้าสู่ระบบและป้อนรหัสผ่าน ข้อจำกัดในการใช้งานระบบของผู้ใช้ (เช่น ผู้ใช้สามารถป้อนรหัสผ่านได้ไม่เกิน 3 ครั้ง) การป้องกันไวรัสคอมพิวเตอร์ การใช้ประโยชน์จากเทคนิควิทยาการเข้ารหัสลับ (cryptographical techniques) การทำ log file หรือเก็บ historical data การจำกัดการสื่อสารระหว่างบางพื้นที่ของโปรแกรม การตรวจสอบความสมบูรณ์ของข้อมูล (เช่น เมื่อมีการ UPDATE ข้อมูล ต้องตรวจสอบด้วยว่ามีการ UPDATE ข้อมูลทุกตารางที่สัมพันธ์กันหรือไม่) และการเข้าถึงข้อมูลแบบอ่านอย่างเดียว เป็นต้น โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสความมั่นคง (Security_code) รายละเอียดดังตารางที่ 3.42

ตารางที่ 3.42 รายละเอียดในตาราง Sec03_Security

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Security_code (PK)	ตัวอักษร	20	รหัสความมั่นคง
Security_description	ตัวอักษร	200	รายละเอียดความมั่นคง
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec03_Software_Quality_Factor

ตาราง Sec03_Software_Quality_Factor ใช้เก็บรายละเอียดของคุณลักษณะ (attribute) ของซอฟต์แวร์ที่ทำให้ซอฟต์แวร์มีความเหมาะสมในการใช้งาน ได้แก่ ความถูกต้อง (correct) ไม่มี ความคลุมเครือ (unambiguous) ความสมบูรณ์ (complete) ความต้องกัน (consistent) ความคงที่ (stability) สามารถทวนสอบได้ (verification) สามารถแก้ไขได้ (modification) และสามารถติดตามได้ (traceable) ในหัวข้อนี้ จะไม่รวมถึงคุณลักษณะหลัก ๆ ที่เกี่ยวกับคุณภาพ (เช่น ความเชื่อถือได้ ความสามารถในการบำรุงรักษา ความปลอดภัย) โดยอาจจะแสดงถึงคุณลักษณะเหล่านี้ในรูปแบบของเมตริกซ์ (metrics) ได้ โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสปัจจัยที่มีผลต่อคุณภาพ (Soft_Qua_Fac_code) รายละเอียดดังตารางที่ 3.43

ตารางที่ 3.43 รายละเอียดในตาราง Sec03_Software_Quality_Factor

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Soft_Qua_Fac_code (PK)	ตัวอักษร	20	รหัสปัจจัยที่มีผลต่อคุณภาพ
Soft_Qua_Fac_description	ตัวอักษร	200	รายละเอียดปัจจัยที่มีผลต่อคุณภาพ
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec03_Standard_Compliance

ตาราง Sec03_Standard_Compliance ใช้เก็บรายละเอียดกฎระเบียบหรือมาตรฐานที่มีใช้อยู่แล้วในองค์กร ซึ่งจะนำมาใช้ในการพัฒนาซอฟต์แวร์ เช่น รูปแบบรายงาน การตั้งชื่อข้อมูล เป็นต้น ถ้ามีการใช้มาตรฐานอื่น ๆ ควรที่จะระบุไว้ในส่วนนี้ด้วย โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสมาตรฐานที่ใช้ในองค์กร (Standard_Comp_code) รายละเอียดดังตารางที่ 3.44

ตารางที่ 3.44 รายละเอียดในตาราง Sec03_Standard_Compliance

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Standard_Comp_code (PK)	ตัวอักษร	20	รหัสมาตรฐานที่มีใช้ในองค์กร
Standard_Comp_description	ตัวอักษร	200	รายละเอียดมาตรฐานที่มีใช้ในองค์กร
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec03_Testcase

ตาราง Sec03_Testcase ใช้เก็บรายละเอียดของ โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) รหัสความต้องการด้านหน้าที่ (Func_Req_code) และรหัสกรณีทดสอบ (Testcase_no) รายละเอียดดังตารางที่ 3.45

ตารางที่ 3.45 รายละเอียดในตาราง Sec03_Testcase

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Func_Req_code (PK)	ตัวอักษร	20	รหัสความต้องการด้านหน้าที่
Testcase_no (PK)	ตัวอักษร	20	รหัสกรณีทดสอบ
Testcase_item	ตัวอักษร	200	รายละเอียดของกรณีทดสอบ

Precondition	ตัวอักษร	200	เงื่อนไขก่อนใช้กรณีทดสอบ
--------------	----------	-----	--------------------------

ตารางที่ 3.46 รายละเอียดในตาราง Sec03_Testcase (ต่อ)

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Postcondition	ตัวอักษร	200	เงื่อนไขหลังใช้กรณีทดสอบ
Expectedoutput	ตัวอักษร	200	ผลลัพธ์ที่ต้องการ
Passfail	ตัวอักษร	1	สถานะผ่าน/ไม่ผ่าน
Assumptionconstraints	ตัวอักษร	200	ข้อสมมุติฐานและข้อจำกัด
Dependencies	ตัวอักษร	200	การขึ้นตรงต่อกัน
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec03_Testcase_Input

ตาราง Sec03_Testcase_Input ใช้เก็บรายละเอียดของกรณีทดสอบที่เก็บมาจากนักวิเคราะห์ระบบ โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) รหัสความต้องการด้านหน้าที่ (Func_Req_code) รหัสกรณีทดสอบ (Testcase_no) และรหัสข้อมูลเข้า (Input_no) รายละเอียดดังตารางที่ 3.46

ตารางที่ 3.47 รายละเอียดในตาราง Sec03_Testcase_Input

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Func_Req_code (PK)	ตัวอักษร	20	รหัสความต้องการด้านหน้าที่
Testcase_no (PK)	ตัวอักษร	20	รหัสกรณีทดสอบ
Input_no (PK)	ตัวอักษร	20	รหัสข้อมูลเข้า

Input_Specification	ตัวอักษร	200	รายละเอียดข้อมูลเข้า
---------------------	----------	-----	----------------------

ตาราง Sec03_Training_Related_Req

ตาราง Sec03_Training_Related_Req ใช้เก็บรายละเอียดของความต้องการในการฝึกอบรมสำหรับบุคคลที่เกี่ยวข้องในระบบ แบ่งเป็น 2 ประเภทคือ การฝึกอบรมสำหรับบุคลากรในทีมพัฒนาซอฟต์แวร์และการฝึกอบรมสำหรับผู้ใช้หรือลูกค้า การฝึกอบรมสำหรับบุคลากรในทีมพัฒนา แบ่งการฝึกอบรมเป็น 2 ส่วน ได้แก่ ด้านเทคนิค (เช่น การใช้โปรแกรม การใช้เครื่องมือเพื่อทดสอบ) และด้านไม่ใช่เทคนิค (เช่น การจัดการ procedures การควบคุมในเรื่องเอกสาร) การฝึกอบรมสำหรับผู้ใช้หรือลูกค้า มีการอบรมเกี่ยวกับการติดตั้งซอฟต์แวร์ การใช้ซอฟต์แวร์ การบำรุงรักษาซอฟต์แวร์ และการเปลี่ยนแปลงปรับปรุงซอฟต์แวร์ โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสความต้องการด้านการฝึกอบรมบุคคลที่เกี่ยวข้อง (Train_Related_Req_code) รายละเอียดดังตารางที่ 3.47

ตารางที่ 3.48 รายละเอียดในตาราง Sec03_Training_Related_Req

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Train_Related_Req_code (PK)	ตัวอักษร	20	รหัสความต้องการด้านการฝึกอบรมบุคคลที่เกี่ยวข้อง
Train_Related_Req_description	ตัวอักษร	200	รายละเอียดความต้องการด้านการฝึกอบรมบุคคลที่เกี่ยวข้อง
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec03_Verification_Req

ตาราง Sec03_Verification_Req ใช้เก็บรายละเอียดวิธีในการตรวจสอบความถูกต้องของซอฟต์แวร์ โดยอาจใช้วิธีการจำลองระบบ (simulation/emulation) การทดสอบในสภาพแวดล้อมจริง โดยใช้ข้อมูลที่จำลองขึ้นมาหรือข้อมูลจริง และทำการทดสอบการต่อประสานไปยังระบบอื่นๆ ภายนอก โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสความต้องการด้านการทดสอบ (Verification_Req_code) รายละเอียดดังตารางที่ 3.48

ตารางที่ 3.49 รายละเอียดในตาราง Sec03_Verification_Req

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Verification_Req_code (PK)	ตัวอักษร	20	รหัสความต้องการด้านการทดสอบ
Verification_Req_description	ตัวอักษร	200	รายละเอียดความต้องการด้านการทดสอบ
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec04_Appendix

ตาราง Sec04_Appendix ใช้เก็บรายละเอียดของภาคผนวก เป็นส่วนที่บรรจุรายละเอียดที่ไม่ได้เขียนอยู่ใน SRS หรือเป็นส่วนที่อธิบายเนื้อหาเพิ่มเติมใน SRS อาจแบ่งแยกเป็นส่วนๆ ได้ เช่น Template ต่างๆ คำอธิบายของการศึกษาการวิเคราะห์ค่าใช้จ่าย หรือผลลัพธ์ของการสำรวจผู้ใช้ ข้อมูลที่ช่วยให้ผู้อ่านอ่าน SRS ได้เข้าใจมากขึ้น เป็นต้น โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code)

รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสภาคผนวก (Appendix_code) รายละเอียดดังตารางที่ 3.49

ตารางที่ 3.50 รายละเอียดในตาราง Sec04_Appendix

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Appendix_code (PK)	ตัวอักษร	20	รหัสภาคผนวก
Appendix_description	ตัวอักษร	200	รายละเอียดภาคผนวก
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec04_Index

ตาราง Sec04_Index ใช้เก็บรายละเอียดของดัชนี เพื่อสะดวกในการค้นหาคำต่าง ๆ ที่ต้องการได้อย่างรวดเร็ว โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสดัชนี (Index_code) รายละเอียดดังตารางที่ 3.50

ตารางที่ 3.51 รายละเอียดในตาราง Sec04_Index

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Index_code (PK)	ตัวอักษร	20	รหัสดัชนี
Index_description	ตัวอักษร	200	รายละเอียดดัชนี

Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec04_Note

ตาราง Sec04_Note ใช้เก็บรายละเอียดต่าง ๆ ที่ต้องการเพิ่มเติมใน SRS เพื่อให้เกิดความเข้าใจมากขึ้น โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสบันทึก (Note_code) รายละเอียดดังตารางที่ 3.51

ตารางที่ 3.52 รายละเอียดในตาราง Sec04_Note

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Note_code (PK)	ตัวอักษร	20	รหัสบันทึก
Note_description	ตัวอักษร	200	รายละเอียดบันทึก
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

ตาราง Sec04_Table_Of_Content

ตาราง Sec04_Table_Of_Content ใช้เก็บรายละเอียดของสารบัญ เพื่อแสดงถึงหัวข้อต่าง ๆ ที่มีอยู่ใน SRS ที่สร้างขึ้น โดยมีคีย์หลัก คือ รหัสลูกค้า (Cust_code) รหัสrelease (Release_code) รหัสโครงการ (Project_code) รหัสข้อกำหนดความต้องการซอฟต์แวร์ (SRS_code) และรหัสสารบัญ (Tab_Cont_code) รายละเอียดดังตารางที่ 3.52

ตารางที่ 3.53 รายละเอียดในตาราง Sec04_Table_Of_Content

ชื่อเขตข้อมูล	ชนิดของข้อมูล	ขนาด	หมายเหตุ
Cust_code (PK)	ตัวอักษร	20	รหัสลูกค้า
Release_code (PK)	ตัวอักษร	10	รหัสrelease
Project_code (PK)	ตัวอักษร	20	รหัสโครงการ
SRS_code (PK)	ตัวอักษร	20	รหัสข้อกำหนดความต้องการ
Tab_Cont_code (PK)	ตัวอักษร	20	รหัสสารบัญ
Tab_Cont_description	ตัวอักษร	200	รายละเอียดสารบัญ
Upd_by	ตัวอักษร	15	ผู้ปรับปรุงข้อมูล
Upd_date	วันที่	-	วันที่ปรับปรุงข้อมูล

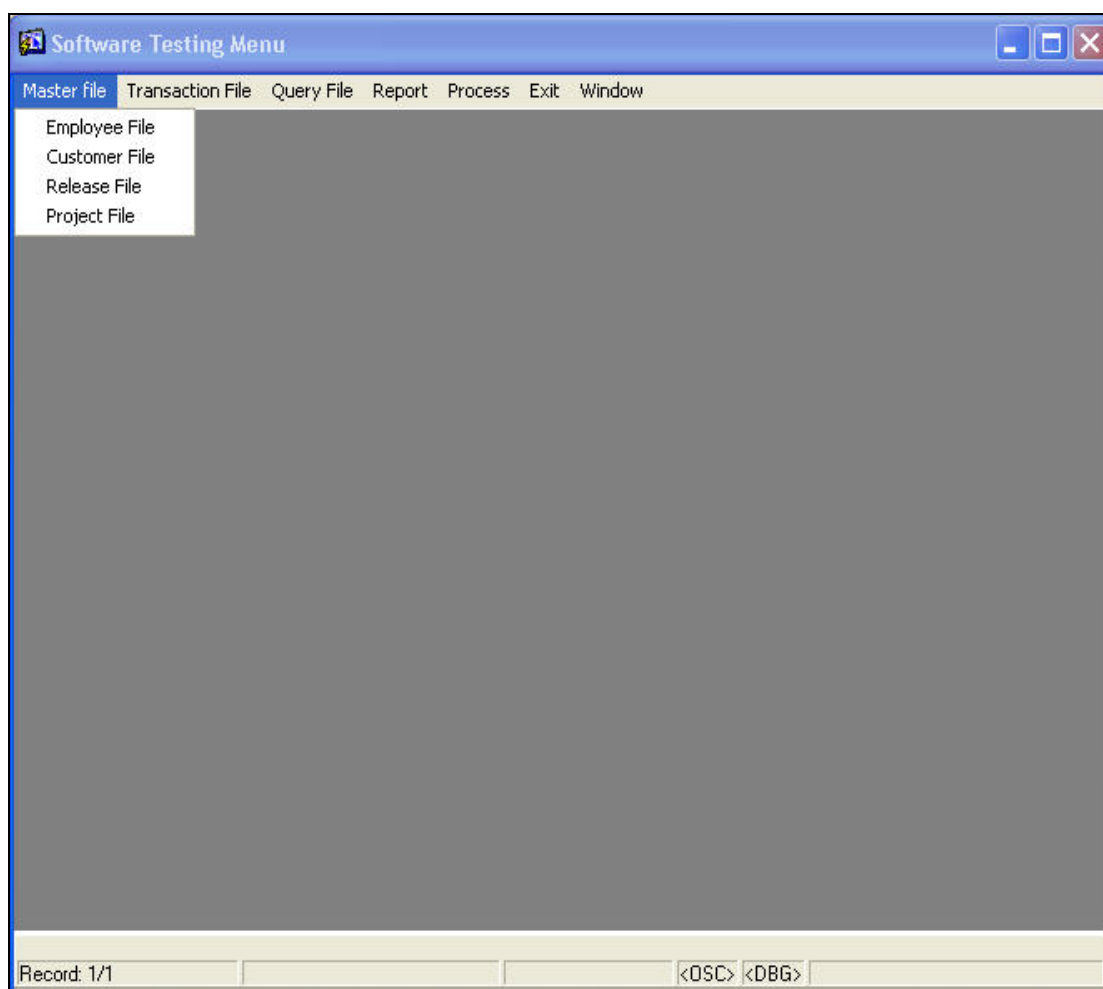
3.4 การออกแบบส่วนนำเข้าข้อมูล

ส่วนนำเข้าข้อมูลของเครื่องมือนี้ มีหน้าจอหลัก 7 ส่วนคือ หน้าจอเมนูหลัก หน้าจอบันทึก รายละเอียดพนักงาน หน้าจอบันทึกรายละเอียดลูกค้า หน้าจอบันทึกรายละเอียด Release หน้าจอ บันทึกรายละเอียดโครงการ หน้าจอบันทึกข้อกำหนดความต้องการซอฟต์แวร์ และหน้าจอบันทึกกรณีทดสอบ

หน้าจอเมนูหลัก

เมื่อเริ่มต้นโปรแกรมทุกครั้ง ผู้ใช้งานจะเข้าสู่หน้าจอเมนูหลัก ดังรูปที่ 3.5 ซึ่งประกอบไปด้วยส่วนต่าง ๆ คือ

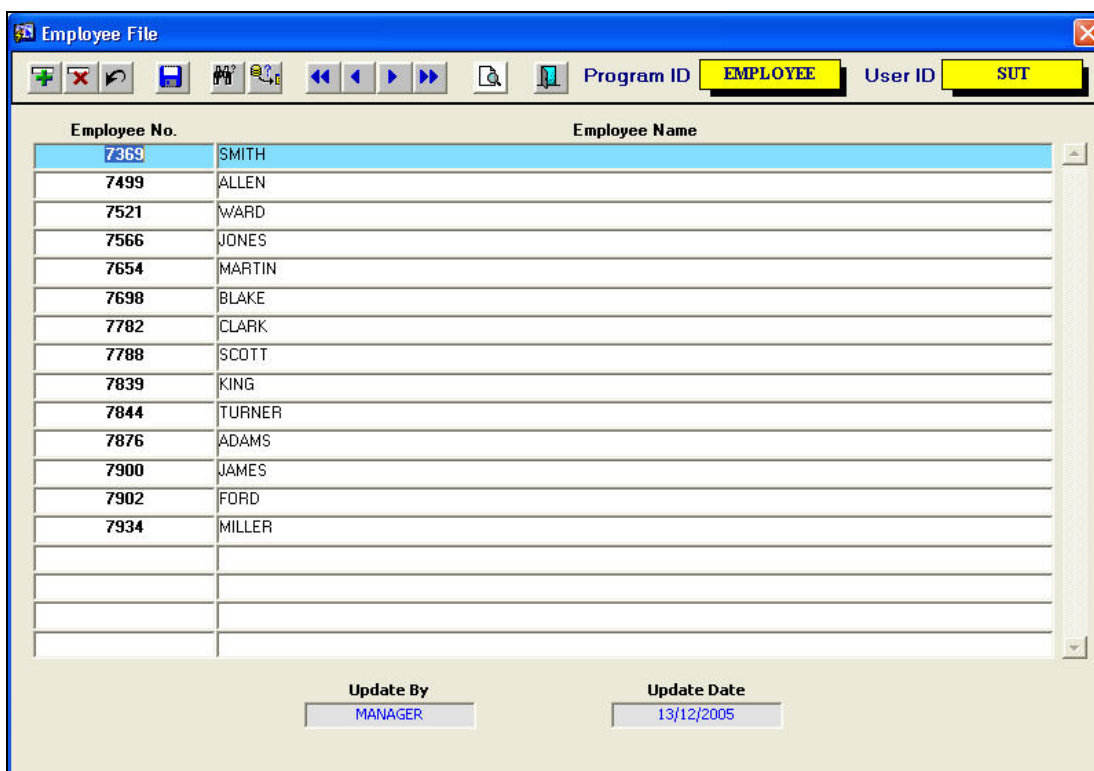
- Master File
- Transaction File
- Query File
- Report
- Exit
- Window



รูปที่ 3.5 หน้าจอเมนูหลัก

หน้าจอบันทึกรายละเอียดพนักงาน

หน้าจอนี้เป็นหน้าจอที่รับข้อมูลรายละเอียดของพนักงาน ได้แก่ รหัสพนักงาน และชื่อพนักงาน ดังแสดงในรูปที่ 3.6



Employee No.	Employee Name
7369	SMITH
7499	ALLEN
7521	WARD
7566	JONES
7654	MARTIN
7698	BLAKE
7782	CLARK
7788	SCOTT
7839	KING
7844	TURNER
7876	ADAMS
7900	JAMES
7902	FORD
7934	MILLER

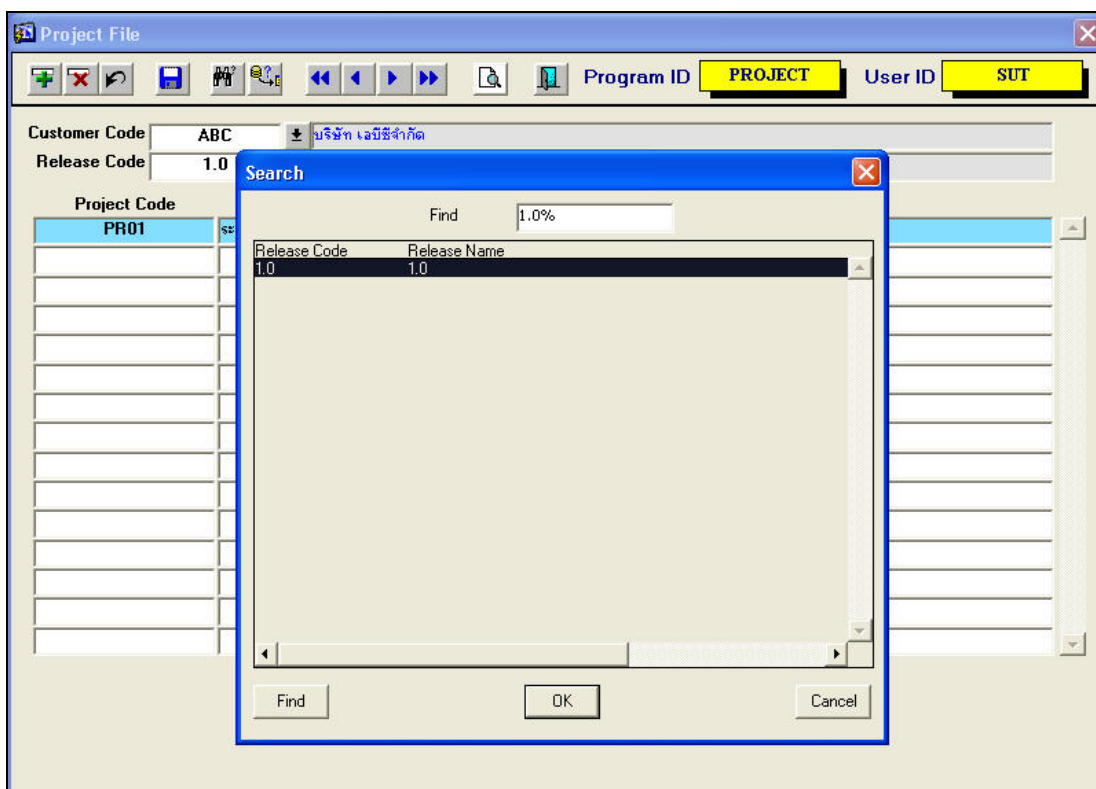
Update By: MANAGER Update Date: 13/12/2005

รูปที่ 3.6 หน้าจอบันทึกรายละเอียดพนักงาน

สำหรับผู้ปรับปรุงล่าสุด และวันที่ปรับปรุงล่าสุด โปรแกรมจะทำการบันทึกให้อัตโนมัติ โดยจะนำข้อมูลจาก User Login และ วันที่ปัจจุบัน มาทำการบันทึก

หน้าจอบันทึกรายละเอียดลูกค้า

หน้าจอนี้เป็นหน้าจอที่จะทำการรับข้อมูลรายละเอียดของลูกค้า ได้แก่ รหัสลูกค้า และชื่อลูกค้า ดังแสดงในรูปที่ 3.7



รูปที่ 3.9 หน้าจอบันทึกรายละเอียดโครงการ

หน้าจอบันทึกข้อกำหนดความต้องการซอฟต์แวร์

หน้าจอนี้เป็นหน้าจอที่รับข้อมูลรายละเอียดของข้อกำหนดความต้องการซอฟต์แวร์ ได้แก่ รหัสลูกค้า รหัสrelease รหัสโครงการ รหัสข้อกำหนดความต้องการซอฟต์แวร์ รายละเอียดข้อกำหนดความต้องการซอฟต์แวร์ เอกสารอ้างอิง ผู้ปรับปรุง และวันที่ปรับปรุง สำหรับในส่วนของการบันทึกรายละเอียดข้อกำหนดความต้องการซอฟต์แวร์นั้นได้แบ่งออกเป็น 4 ส่วนด้วยกัน ดังนี้คือ

ส่วนที่ 1 เป็นส่วนของบทนำ (Introduction) ประกอบไปด้วย จุดประสงค์ (Purpose) ขอบเขต (Scope) คำจำกัดความ (Definitions) รหัสพจน์ (Acronyms) คำย่อ (Abbreviations) เอกสารอ้างอิง (Referemces) และภาพรวม (Overview)

ส่วนที่ 2 เป็นส่วนของรายละเอียดโดยรวม (Oveerall description) ประกอบไปด้วย การต่อประสานกับผู้ใช้ (User Interfaces) การต่อประสานกับฮาร์ดแวร์ (Hardware Interfaces) การต่อประสานกับซอฟต์แวร์ (Software Interfaces) การต่อประสานกับระบบสื่อสาร (Communications

Interfaces) ข้อจำกัดของหน่วยความจำ (Memory constraints) การดำเนินการ (Operations) ความต้องการด้านการติดตั้งซอฟต์แวร์ (Site adaptation requirements) หน้าที่ของซอฟต์แวร์ (Product functions) ข้อจำกัดในการเตรียมกระบวนการในการพิจารณาซอฟต์แวร์ (Qualification provisions) ความสามารถในการติดตามความต้องการ (Requirements traceability) ข้อจำกัด (Constraints) ข้อสันนิษฐานและการขึ้นต่อกัน (Assumptions and dependencies) และการแบ่งสัดส่วนความต้องการ (Apportioning of requirements)

ส่วนที่ 3 เป็นส่วนของความต้องการที่เฉพาะเจาะจง (Specific requirements) ประกอบด้วย การต่อประสานภายนอก (External interface) ความต้องการด้านหน้าที่ (Functional requirements) ความต้องการด้านประสิทธิภาพ (Performance requirements) ความต้องการทางด้านฐานข้อมูลเชิงตรรกะ (Logical database requirements) ความต้องการทางด้านการดำเนินการ (Operational requirements) ความต้องการด้านการทวนสอบ (Verification requirements) ความต้องการด้านการยอมรับ (Acceptance requirements) ความต้องการด้านเอกสาร (Documentation requirements) ความต้องการด้านความปลอดภัย (Safety requirements) ความต้องการด้านการใช้ทรัพยากรฮาร์ดแวร์คอมพิวเตอร์ให้เกิดประโยชน์สูงสุด (Computer hardware resource utilization requirements) ความต้องการด้านระบบสื่อสารคอมพิวเตอร์ (Computer communications requirements) ปัจจัยที่มีผลต่อคุณภาพของซอฟต์แวร์ (Software quality factors) ความต้องการด้านการฝึกอบรมของบุคคลที่เกี่ยวข้อง (Training-relates requirements) ความต้องการทางด้านการเตรียมขั้นตอนต่าง ๆ ที่ใช้ในการพัฒนาซอฟต์แวร์ (Logistics-relates requirements) ความต้องการทางด้านการบรรจุภัณฑ์ซอฟต์แวร์ (Packaging requirements) ข้อจำกัดในการใช้งานทางด้านซอฟต์แวร์ (Implementation constraints) มาตรฐานที่มีใช้ภายในองค์กร (Standard compliance) ความเชื่อถือได้ (Reliability) สภาพของความพร้อมใช้งาน (Availability) ความมั่นคง (Security) ความสามารถในการบำรุงรักษา (Maintainability) และความสามารถในการใช้ได้หลายระบบ (Portability)

ส่วนที่ 4 เป็นส่วนของข้อมูลสนับสนุน (Supporting information) ประกอบไปด้วย สารบัญ (Table of contents) บันทึก (Notes) ภาคผนวก (Appendixes) และดัชนี (Index) ดังแสดงในรูปที่ 3.10

Software Requirement Specification File

Program ID: **TEST** User ID: **SUT**

Customer Code: **ABC** บริษัท เอ็มซีจำกัด

Release Code: **1.0** 1.0

Project Code: **PR01** ระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติ

Srs Code: **SRS01** SRS ของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติของบริษัท เอ็มซี Ref No:

Prepare By: **NARIS** Prepare Date: **01/11/2005**

SECTION01 | SECTION02(1/2) | SECTION02(2/2) | SECTION03(1/3) | SECTION03(2/3) | SECTION03(3/3) | SECTION04

Abbreviation

01	CID - Customer Identification Number
02	PID - Product Identification Number

Overview

Scope

01	ข้อกำหนดฉบับนี้ ใช้เป็นพื้นฐานในการออกแบบฐานข้อมูล
----	----------------------------------------------------

Acronyms

01	IEEE - Institute of Electrical and Electronic Engineering
02	SDS - Software Design Specification

Purpose

01	SRS ฉบับนี้อธิบายถึงความต้องการทางด้านหน้าที่และความต้องการทางด้านประสิทธิภาพในระบบย่อย
----	-----------------------------------------------------------------------------------------

Definition

01	ลูกค้า - เป็นผู้ที่ต้องการซอฟต์แวร์ที่ได้รับการพัฒนาแล้ว
02	ผู้ใช้ - เป็นผู้ที่ใช้ประโยชน์จากซอฟต์แวร์ที่พัฒนาสมบูรณ์แล้ว ซึ่งอาจเป็นคนเดียวกับลูกค้าก็ได้

Reference

01	IEEE Std 610.12-1990, IEEE Standard Glossary for Software Engineering
----	-----------------------------------------------------------------------

รูปที่ 3.10 หน้าจอบันทึกข้อกำหนดความต้องการซอฟต์แวร์

Oracle Forms Runtime - [WINDOW1]

Action Edit Query Block Record Field Window Help

Customer Code ABC V บัณฑิต เอ็ม ซี จำกัด

Release Code 1.0 V 1.0

Project Code PR01 V ระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติ

Srs Code SR501 SRS ของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติของบริษัท เอ็ม ซี จำกัด

Prepare By NARIS Prepare Date 01/11/2005

Ref No

SECTION01 SECTION02(1/2) SECTION02(2/2) SECTION03(1/3) SECTION03(2/3) SECTION03(3/3) SE

Apport Requirement	Constraint Code	Operation Code
Assump Depend	Hard Code	Prod Func Code
Comm Code	Mem	Qua Prov Code

Record: 1/1

รูปที่ 3.11 หน้าจอบันทึกข้อกำหนดความต้องการซอฟต์แวร์

Oracle Forms Runtime - [WINDOW1]

Action Edit Query Block Record Field Window Help

Customer Code ABC V บริษัท เอบีซี จำกัด

Release Code 1.0 V 1.0

Project Code PR01 V ระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติ

Srs Code SR501 SRS ของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติของ บริษัทเอบีซี Ref No

Prepare By NARIS Prepare Date 01/11/2005

SECTION01 SECTION02(1/2) SECTION02(2/2) SECTION03(1/3) SECTION03(2/3) SECTION03(3/3) SE

Req Trace User Char

Site Adap Req User Interface

Soft Code

Record: 1/1

รูปที่ 3.12 หน้าจอบันทึกข้อกำหนดความต้องการซอฟต์แวร์

Oracle Forms Runtime - [WINDOW1]

Action Edit Query Block Record Field Window Help

Customer Code ABC V บริษัท เอ็มซี จำกัด

Release Code 1.0 V 1.0

Project Code PR01 V ระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติ

Srs Code SR501 SRS ของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติของบริษัท เอ็มซี

Ref No

Prepare By NARIS Prepare Date 01/11/2005

SECTION01	SECTION02(1/2)	SECTION02(2/2)	SECTION03(1/3)	SECTION03(2/3)	SECTION03(3/3)	SE
Acceptance Req			Doc Req			Imp Cons
Availability			Env Req			Logic Dat Req
Comp Hard Resource			Ext Interf			Log Related Req

Record: 1/1

รูปที่ 3.13 หน้าจอบันทึกข้อกำหนดความต้องการซอฟต์แวร์

Oracle Forms Runtime - [WINDOW1]

Action Edit Query Block Record Field Window Help

Customer Code ABC V บริษัท เอบีซี จำกัด

Release Code 1.0 V 1.0

Project Code PR01 V ระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติ

Srs Code SR501 SRS ของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติของบริษัท เอบีซี Ref No

Prepare By NARIS Prepare Date 01/11/2005

SECTION01	SECTION02(1/2)	SECTION02(2/2)	SECTION03(1/3)	SECTION03(2/3)	SECTION03(3/3)	SE
Maintainability			Portability		Security	
Package			Reliability		Soft Qua Fac	
Perform Req			Safety Req		Standard Comp	

Record: 1/1

รูปที่ 3.14 หน้าจอบันทึกข้อกำหนดความต้องการซอฟต์แวร์

Oracle Forms Runtime - [WINDOW1]

Action Edit Query Block Record Field Window Help

Customer Code ABC V บริษัท เอบีซี จำกัด

Release Code 1.0 V 1.0

Project Code PR01 V ระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติ

Srs Code SR501 SRS ของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติของบริษัท เอบีซี Ref No

Prepare By NARIS Prepare Date 01/11/2005

SECTION01 SECTION02(1/2) SECTION02(2/2) SECTION03(1/3) SECTION03(2/3) SECTION03(3/3) SE

Train Related Req

Verification Req

Func Req

Condition

Condition

Condition

Condition

Condition

Condition

Record: 1/1

รูปที่ 3.15 หน้าจอบันทึกข้อกำหนดความต้องการซอฟต์แวร์

Oracle Forms Runtime - [WINDOW1]

Action Edit Query Block Record Field Window Help

Customer Code ABC V บัณฑิต เอ็ม ซี จำกัด

Release Code 1.0 V 1.0

Project Code PR01 V ระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติ

Srs Code SR501 SRS ของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติของบริษัท เอ็ม ซี จำกัด

Prepare By NARIS Prepare Date 01/11/2005

Ref No

SECTION02(1/2) SECTION02(2/2) SECTION03(1/3) SECTION03(2/3) SECTION03(3/3) SECTION04

Appendix

Note

Index

Tab Cont

Record: 1/1

รูปที่ 3.16 หน้าจอบันทึกข้อกำหนดความต้องการซอฟต์แวร์

หน้าจอบันทึกข้อมูลกรณีทดสอบ

หน้าจอนี้เป็นหน้าจอที่รับข้อมูลรายละเอียดกรณีทดสอบ โดยกรณีทดสอบแต่ละกรณีจะมีความสัมพันธ์กับความต้องการด้านหน้าที่ (Functional requirements) ได้แก่ รหัสกรณีทดสอบ รายละเอียดของกรณีทดสอบ เงื่อนไขก่อนใช้กรณีทดสอบ เงื่อนไขหลังใช้กรณีทดสอบ รหัสข้อมูลเข้า รายละเอียดข้อมูลเข้า ผลลัพธ์ที่ต้องการ สถานะผ่าน/ไม่ผ่าน ข้อสมมุติฐานและข้อจำกัด และ Dependencies ดังแสดงในรูปที่ 3.17

The screenshot shows a 'Testcase' window with the following fields and sections:

- Testcase No:** A text box containing 'TC20060301'.
- Testcase Item:** A text box with the placeholder 'Brief description of the test items exercised by this test case'.
- Pre-condition:** A text box with the placeholder 'Conditions must be established before the test'.
- Post-condition:** A text box with the placeholder 'Conditions must be established after the test'.
- Input No:** A list box containing 'INT01' and 'INT02'.
- Input Specification:** A text box with the placeholder 'Bullet items to describe each input required for this test case 01'.
- Expected output:** A text box with the placeholder 'Bullet items to describe each expected output for this test case'.
- Pass/Fail Criteria:** Radio buttons for 'PASS' (selected) and 'FAIL'.
- Assumption & Constraints:** A text box with the placeholder 'Bullet items'.
- Dependencies:** A text box with the placeholder 'List all the test cases depend on or are depend by this test case'.
- OK:** A button at the bottom center.

รูปที่ 3.17 หน้าจอบันทึกกรณีทดสอบ

บทที่ 4

การทดสอบและอภิปรายผล

การพัฒนาเครื่องมือสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ โดยผลการทดสอบการทำงานของเครื่องมือ มีรายละเอียดดังนี้ หัวข้อ 4.1 อธิบายถึงสภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือสร้างกรณีทดสอบ หัวข้อ 4.2 อธิบายถึงโครงสร้างของเครื่องมือ หัวข้อ 4.3 อธิบายถึงกรณีทดสอบที่ใช้ทดสอบเครื่องมือ หัวข้อ 4.4 อธิบายถึงขั้นตอนการทดสอบ หัวข้อ 4.5 อธิบายถึงผลการทดสอบโปรแกรม และหัวข้อ 4.6 เป็นการอภิปรายสรุป

4.1 สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือสร้างกรณีทดสอบ

เครื่องคอมพิวเตอร์ที่ใช้ในการพัฒนามีรายละเอียดดังนี้

เครื่องให้บริการ (Server)

ฮาร์ดแวร์

- คอมพิวเตอร์แบบพีซี Pentium IV 3.2 กิกะเฮิร์ต
- หน่วยความจำ 512 เมกกะไบต์
- ฮาร์ดดิสก์ 80 กิกะไบต์

ซอฟต์แวร์

- ระบบปฏิบัติการ ไมโครซอฟต์วินโดวส์ เอ็กซ์พี
- ระบบฐานข้อมูล ออราเคิล 10gXE

เครื่องรับบริการ (Client)

ฮาร์ดแวร์

- คอมพิวเตอร์แบบพีซี Pentium IV 3.2 กิกะเฮิร์ต
- หน่วยความจำ 512 เมกกะไบต์
- ฮาร์ดดิสก์ 80 กิกะไบต์

ซอฟต์แวร์

- ระบบปฏิบัติการ ไมโครซอฟต์วินโดวส์ เอ็กซ์พี
- เครื่องมือพัฒนา ออราเคิลดีเวลอปเปอร์ 6ไอ (Oracle Developer 6i)

4.2 โครงสร้างของเครื่องมือ

โปรแกรมที่พัฒนานี้ประกอบด้วยโมดูลหลัก ซึ่งแต่ละโมดูลจะถูกเรียกจากหน้าจอเมนูหลักของโปรแกรม คือโมดูลควบคุมหลัก (Main controller) ซึ่งเป็นโมดูลที่ให้ผู้ใช้งานเลือกการทำงานกับโปรแกรม ประกอบด้วยโมดูลย่อยต่าง ๆ ดังนี้

- 1) ส่วนของแฟ้มข้อมูลหลัก (Master File) เป็นส่วนที่รับข้อมูลพื้นฐานเกี่ยวกับการบันทึกโครงการทดสอบจากผู้ทดสอบ และบันทึกข้อมูลลงในฐานข้อมูล ผู้ทดสอบสามารถเพิ่ม ลด หรือแก้ไขข้อมูลโครงการทดสอบได้ และมีส่วนของการแสดงข้อมูลความก้าวหน้าโดยรวมของแต่ละโครงการทดสอบที่บันทึกไว้ในฐานข้อมูลด้วย
- 2) ส่วนของแฟ้มข้อมูลรายวัน (Transaction File) ทำหน้าที่รับข้อมูลรายละเอียดของข้อกำหนดความต้องการซอฟต์แวร์ เพื่อใช้สร้างกรณีทดสอบ ผู้ทดสอบสามารถเพิ่ม ลด หรือแก้ไขข้อมูลกรณีทดสอบได้ หรือเพิ่มรายการทดสอบที่ต้องการลงในฐานข้อมูลได้
- 3) ส่วนของการสอบถาม (Query File) เป็นส่วนที่จะแสดงข้อมูลเกี่ยวกับข้อกำหนดความต้องการซอฟต์แวร์และกรณีทดสอบที่ได้บันทึกลงฐานข้อมูลไว้แล้ว ในส่วนนี้ผู้ทดสอบไม่สามารถแก้ไขเพิ่มเติมหรือลบข้อมูลได้
- 4) ส่วนของรายงาน (Report) เป็นส่วนที่นำข้อมูลที่เก็บไว้ในฐานข้อมูลกรณีทดสอบมาออกรายงานให้กับผู้ทดสอบ รายงานที่สร้างขึ้นโดยส่วนนี้ได้แก่ รายงานกรณีทดสอบเป็นรายงานที่เครื่องมือออก ให้ผู้ทดสอบเพื่อนำไปใช้ในการทดสอบตามรายการทดสอบที่ระบุไว้ เครื่องมือจะสร้างรายงานนี้ให้กับผู้ทดสอบเมื่อผู้ทดสอบบันทึกข้อมูลการสร้างกรณีทดสอบเสร็จสิ้น

4.3 กรณีทดสอบที่ใช้ทดสอบเครื่องมือ

เป็นการสร้างกรณีทดสอบสำหรับระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติ การออกแบบและสร้างกรณีทดสอบเพื่อใช้ทดสอบระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติมีขั้นตอนดังนี้

- ยอมรับใบสั่งซื้อสินค้าที่ถูกต้องจากลูกค้า
- แจ้งราคาสินค้าให้ถูกต้อง (รวมค่าภาษีและค่าส่งทางไปรษณีย์)
- ยืนยันสถานะบัตรเครดิตของลูกค้า และมีการตรวจสอบบัตรเครดิตให้ถูกต้อง
- ส่งข้อมูลเกี่ยวกับการสั่งซื้อสินค้าไปยังแผนกส่งของ
- ส่งใบเสร็จเก็บเงินไปยังแผนกลูกค้า
- ปรับแฟ้มข้อมูลของลูกค้าให้ทันสมัย

- เปิดเพิ่มข้อมูลลูกค้าใหม่
- เพิ่มข้อมูลลูกค้าไปยังบัญชีรายชื่อผู้ที่สั่งซื้อสินค้าทางไปรษณีย์
- ทำสมุดจดรายการเปลี่ยนแปลงในแต่ละรายการ
- บันทึกรายการเปลี่ยนแปลงในแต่ละรายการลงบนเทป

4.4 ขั้นตอนการทดสอบ

- 1) เรียกโปรแกรม สร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ เพื่อทำการบันทึกข้อมูล
- 2) ทำการบันทึกข้อมูลในส่วนของการเพิ่มข้อมูลหลักซึ่งประกอบไปด้วย หน้าจอบันทึกรายละเอียดพนักงาน, หน้าจอบันทึกรายละเอียดลูกค้า, หน้าจอบันทึกรายละเอียด Release และ หน้าจอบันทึกรายละเอียดโครงการ
- 3) บันทึกข้อมูลในส่วนของการกำหนดความต้องการซอฟต์แวร์ โดยนำข้อมูลที่ได้เก็บรวบรวมมาทำการบันทึกลงในโปรแกรม
- 4) สร้างกรณีทดสอบสำหรับการทดสอบการยอมรับโดยกดปุ่ม เงื่อนไข (Condition) ซึ่งปรากฏอยู่ที่รายการความต้องการด้านหน้าที่ (Functional requirements)
- 5) บันทึกข้อมูลลักษณะเฉพาะ หรือเงื่อนไขของข้อมูลเข้าเพื่อให้เครื่องมือนำไปสร้างกรณีทดสอบ
- 6) เครื่องมือสามารถสร้างเอกสารแสดงกรณีทดสอบที่สร้างขึ้นสำหรับแต่ละรายการทดสอบ
- 7) ผู้ทดสอบสามารถพิมพ์เอกสารตัวแบบในส่วนของการกำหนดความต้องการซอฟต์แวร์เพื่อนำไปใช้ในการเก็บความต้องการได้ และสามารถพิมพ์เอกสารตัวแบบในส่วนของการทดสอบเพื่อนำไปใช้ในการทดสอบ

4.5 ผลการทดสอบโปรแกรม

การทดสอบเริ่มจากผู้ทดสอบต้องพิมพ์เอกสารตัวแบบในส่วนของการกำหนดความต้องการซอฟต์แวร์ เพื่อนำไปใช้ในการเก็บความต้องการ แล้วนำความต้องการที่ได้มาบันทึกลงในระบบโดยใช้โปรแกรมบันทึกข้อกำหนดความต้องการซอฟต์แวร์ โดยก่อนการบันทึกข้อกำหนดความต้องการซอฟต์แวร์ ผู้ทดสอบต้องบันทึกข้อมูลในส่วนของการเพิ่มข้อมูลหลักให้เรียบร้อยแล้ว ซึ่งประกอบไปด้วยโปรแกรม บันทึกรายละเอียดพนักงาน, บันทึกรายละเอียดลูกค้า, บันทึกรายละเอียด Release และ บันทึกรายละเอียดโครงการ หลังจากนั้นผู้ทดสอบจึงทำการสร้างกรณี

ทดสอบ โดยกดปุ่ม เงื่อนไข (Condition) ซึ่งจะปรากฏอยู่ในรายการความต้องการด้านหน้าที่สำหรับการสร้างกรณีทดสอบ ผู้ทดสอบจะต้องทำการวิเคราะห์ความต้องการ และอาศัยหลักการการออกแบบกรณีทดสอบแบบแบล็กบ็อกซ์ โดยใช้เทคนิคการสร้าง Thread Testing ซึ่งเป็นลักษณะของการทดสอบตามงานที่ทำ ในลักษณะของการใช้งานระบบ เพื่อตรวจสอบความต้องการทางด้านธุรกิจ เมื่อผู้ทดสอบทำการวิเคราะห์เรียบร้อยแล้ว จึงนำผลการวิเคราะห์มาบันทึกลงในโปรแกรมบันทึกข้อมูลกรณีทดสอบ

เมื่อผู้ทดสอบทำการบันทึกผลการทดสอบเข้าสู่ระบบเรียบร้อยแล้ว สามารถพิมพ์รายงานกรณีทดสอบต่าง ๆ ได้ โดยรายละเอียดในการสร้างกรณีทดสอบ มีดังนี้ (1) พิมพ์ตัวต้นแบบในส่วนของข้อกำหนดความต้องการซอฟต์แวร์ (2) บันทึกข้อมูลในส่วนของเพิ่มข้อมูลหลัก (3) บันทึกข้อกำหนดความต้องการซอฟต์แวร์ (4) บันทึกกรณีทดสอบของแต่ละความต้องการด้านหน้าที่ และ (5) พิมพ์เอกสารกรณีทดสอบเพื่อนำไปทำการทดสอบ

4.5.1 ขั้นตอนการสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ

ผู้ทดสอบทำการพิมพ์ตัวต้นแบบในส่วนของข้อกำหนดความต้องการซอฟต์แวร์ดังแสดงในรูปที่ 4.1

ข้อโครงการ

เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์:

โดย :

วันที่ :

สารบัญ (Table of contents)

1. บทนำ (Introduction)
 - 1.1. จุดประสงค์ (Purpose)
 - 1.2. ขอบเขต (Scope)
 - 1.3. คำจำกัดความ (Definitions)
 - 1.4. รหัสนิยม (Acronyms)
 - 1.5. คำย่อ (Abbreviations)
 - 1.6. เอกสารอ้างอิง (References)
 - 1.7. ภาพรวม (Overview)
2. รายละเอียดโดยรวม (Overall description)
 - 2.1. แนวคิดหรือภาพรวมของซอฟต์แวร์ (Product perspective)
 - 2.1.1. การต่อประสานกับผู้ใช้ (User Interfaces)
 - 2.1.2. การต่อประสานกับฮาร์ดแวร์ (Hardware Interfaces)
 - 2.1.3. การต่อประสานกับซอฟต์แวร์ (Software Interfaces)
 - 2.1.4. การต่อประสานกับระบบสื่อสาร (Communications Interfaces)
 - 2.1.5. ข้อจำกัดของหน่วยความจำ (Memory constraints)
 - 2.1.6. การดำเนินการ (Operations)
 - 2.1.7. ความต้องการด้านการติดตั้งซอฟต์แวร์ (Site adaptation requirements)
 - 2.2. หน้าที่ของซอฟต์แวร์ (Product functions)
 - 2.3. ข้อกำหนดในการเตรียมกระบวนการในการพิจารณาซอฟต์แวร์ (Qualification provisions)
 - 2.4. ความสามารถในการติดตามความต้องการ (Requirements traceability)
 - 2.5. ข้อจำกัด (Constraints)
 - 2.6. ข้อสันนิษฐานและการขึ้นต่อกัน (Assumptions and dependencies)
 - 2.7. การแบ่งปันส่วนความต้องการ (Apportioning of requirements)

รูปที่ 4.1 ตัวต้นแบบในส่วนของข้อกำหนดความต้องการซอฟต์แวร์

3. ความต้องการที่เฉพาะเจาะจง (Specific requirements)
 - 3.1. การต่อประสานภายนอก (External interface)
 - 3.2. ความต้องการด้านหน้าที่ (Functional requirements)
 - 3.2.1. ความต้องการด้านหน้าที่ที่ 1 (Functional requirements 1)
 - 3.2.1.1. บทนำ (Introduction)
 - 3.2.1.2. ข้อมูลนำเข้า (Inputs)
 - 3.2.1.3. กระบวนการ (Processing)
 - 3.2.1.4. ข้อมูลนำออก (Outputs)
 - 3.2.2. ความต้องการด้านหน้าที่ที่ 2 (Functional requirements 2)
 - ...
 - 3.2.n. ความต้องการด้านหน้าที่ที่ n (Functional requirements n)
 - 3.3. ความต้องการด้านประสิทธิภาพ (Performance requirements)
 - 3.4. ข้อจำกัดในการออกแบบ (Design constraints)
 - 3.4.1. ข้อจำกัดในการใช้งานซอฟต์แวร์ (Implementation constraints)
 - 3.4.2. มาตรฐานที่มีใช้ในองค์กร (Standard compliance)
 - 3.5. ลักษณะประจำ (Attributes)
 - 3.5.1. ความเชื่อถือได้ (Reliability)
 - 3.5.2. สภาพพร้อมใช้งาน (Availability)
 - 3.5.3. ความมั่นคง (Security)
 - 3.5.4. ความสามารถในการบำรุงรักษา (Maintainability)
 - 3.5.5. ความสามารถในการใช้ได้หลายระบบ (Portability)
 - 3.6. ความต้องการอื่นๆ (Other requirements)
 - 3.6.1. ความต้องการด้านฐานข้อมูลเชิงตรรกะ (Logical database requirements)
 - 3.6.2. ความต้องการด้านการดำเนินการ (Operational requirements)
 - 3.6.3. ความต้องการด้านการทดสอบ (Verification requirements)
 - 3.6.4. ความต้องการด้านการยอมรับ (Acceptance requirements)
 - 3.6.5. ความต้องการด้านเอกสาร (Documentation requirements)
 - 3.6.6. ความต้องการด้านความปลอดภัย (Safety requirements)
 - 3.6.7. ความต้องการด้านการใช้ทรัพยากรฮาร์ดแวร์คอมพิวเตอร์ให้เกิดประโยชน์สูงสุด (Computer hardware resource utilization requirements)

รูปที่ 4.1 ตัวต้นแบบในส่วนของข้อกำหนดความต้องการซอฟต์แวร์ (ต่อ)

- 3.6.8. ความต้องการด้านระบบสื่อสารคอมพิวเตอร์ (Computer communications requirements)
- 3.6.9. ปัจจัยที่มีผลต่อคุณภาพของซอฟต์แวร์ (Software quality factors)
- 3.6.10. ความต้องการด้านการฝึกอบรมบุคคลที่เกี่ยวข้อง (Training-relates requirements)
- 3.6.11. ความต้องการด้านการเตรียมขั้นตอนต่างๆที่ใช้ในการพัฒนาซอฟต์แวร์ (Logistics-relates requirements)
- 3.6.12. ความต้องการด้านการบรรจุภัณฑ์ซอฟต์แวร์(Packaging requirements)
- 4. ข้อมูลสนับสนุน (Supporting information)
 - 4.1. สารบัญ (Table of contents)
 - 4.2. บันทึก (Notes)
 - 4.3. ภาคผนวก (Appendixes)
 - 4.4. ดัชนี (Index)

รูปที่ 4.1 ตัวต้นแบบในส่วนของข้อกำหนดความต้องการซอฟต์แวร์ (ต่อ)

- 1) บันทึกข้อมูลในส่วนของแฟ้มข้อมูลหลัก หลังจากพิมพ์ตัวต้นแบบในส่วนของข้อกำหนดความต้องการซอฟต์แวร์แล้ว ผู้ทดสอบต้องบันทึกข้อมูลในส่วนของแฟ้มข้อมูลหลักซึ่งประกอบไปด้วย บันทึกรายละเอียดพนักงาน, บันทึกรายละเอียดลูกค้า, บันทึกรายละเอียดRelease และ บันทึกรายละเอียดโครงการ ดังแสดงในรูปที่ 4.2, รูปที่ 4.3, รูปที่ 4.4 และ รูปที่ 4.5

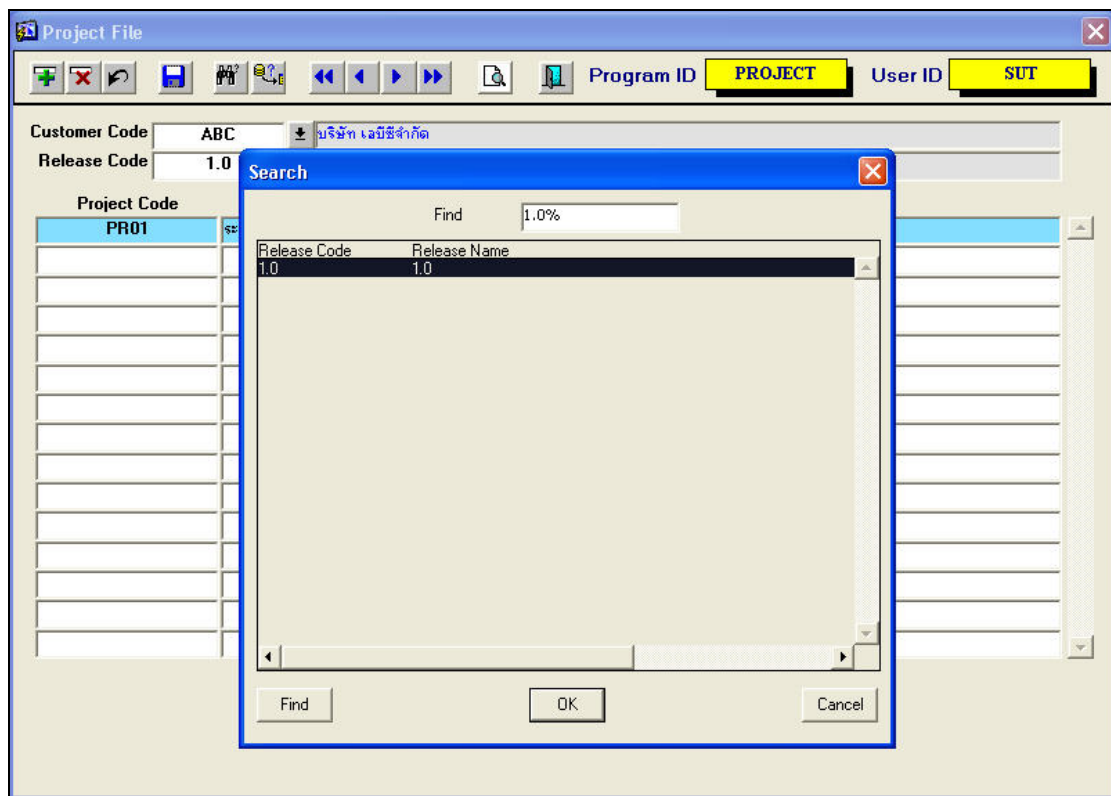
Employee File

Program ID **EMPLOYEE** User ID **SUT**

Employee No.	Employee Name
7369	SMITH
7499	ALLEN
7521	WARD
7566	JONES
7654	MARTIN
7698	BLAKE
7782	CLARK
7788	SCOTT
7839	KING
7844	TURNER
7876	ADAMS
7900	JAMES
7902	FORD
7934	MILLER

Update By: **MANAGER** Update Date: **13/12/2005**

รูปที่ 4.2 บันทึกรายละเอียดพนักงาน



รูปที่ 4.5 บันทึกรายละเอียดโครงการ

- 2) บันทึกข้อกำหนดความต้องการซอฟต์แวร์ หลังจากบันทึกรายละเอียดต่าง ๆ ของรายการในแฟ้มข้อมูลหลักเสร็จเรียบร้อยแล้ว ผู้ทดสอบจึงนำเอาข้อมูลที่ได้จากเอกสารตัวต้นแบบในส่วนของข้อกำหนดความต้องการซอฟต์แวร์ มาบันทึกลงในโปรแกรมบันทึกข้อกำหนดความต้องการซอฟต์แวร์ ดังแสดงในรูปที่ 4.6

The screenshot shows the Oracle Forms Runtime window titled "Oracle Forms Runtime - [WINDOW1]". The menu bar includes Action, Edit, Query, Block, Record, Field, Window, and Help. The toolbar contains various icons for file operations and navigation. The form displays the following data:

Customer Code	ABC	V	บริษัท เอบีซี จำกัด
Release Code	1.0	V	1.0
Project Code	PR01	V	ระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติ
Srs Code	SR501	SRS ของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติของบริษัท เอบีซี	
Prepare By	NARIS	Prepare Date	01/11/2005

Below the data entry fields, there is a section navigation bar with tabs: SECTION01, SECTION02(1/2), SECTION02(2/2), SECTION03(1/3), SECTION03(2/3), and SECTION03(3/3). The current section is SECTION03(3/3). The form is divided into three main sections:

- Train Related Req**: Contains two input fields for training-related requirements.
- Verification Req**: Contains two input fields for verification requirements.
- Func Req**: Contains a table for functional requirements with five rows. Each row has a text input field and a "Condition" button.

At the bottom of the window, the status bar shows "Record: 1/1".

รูปที่ 4.6 บันทึกข้อกำหนดความต้องการซอฟต์แวร์

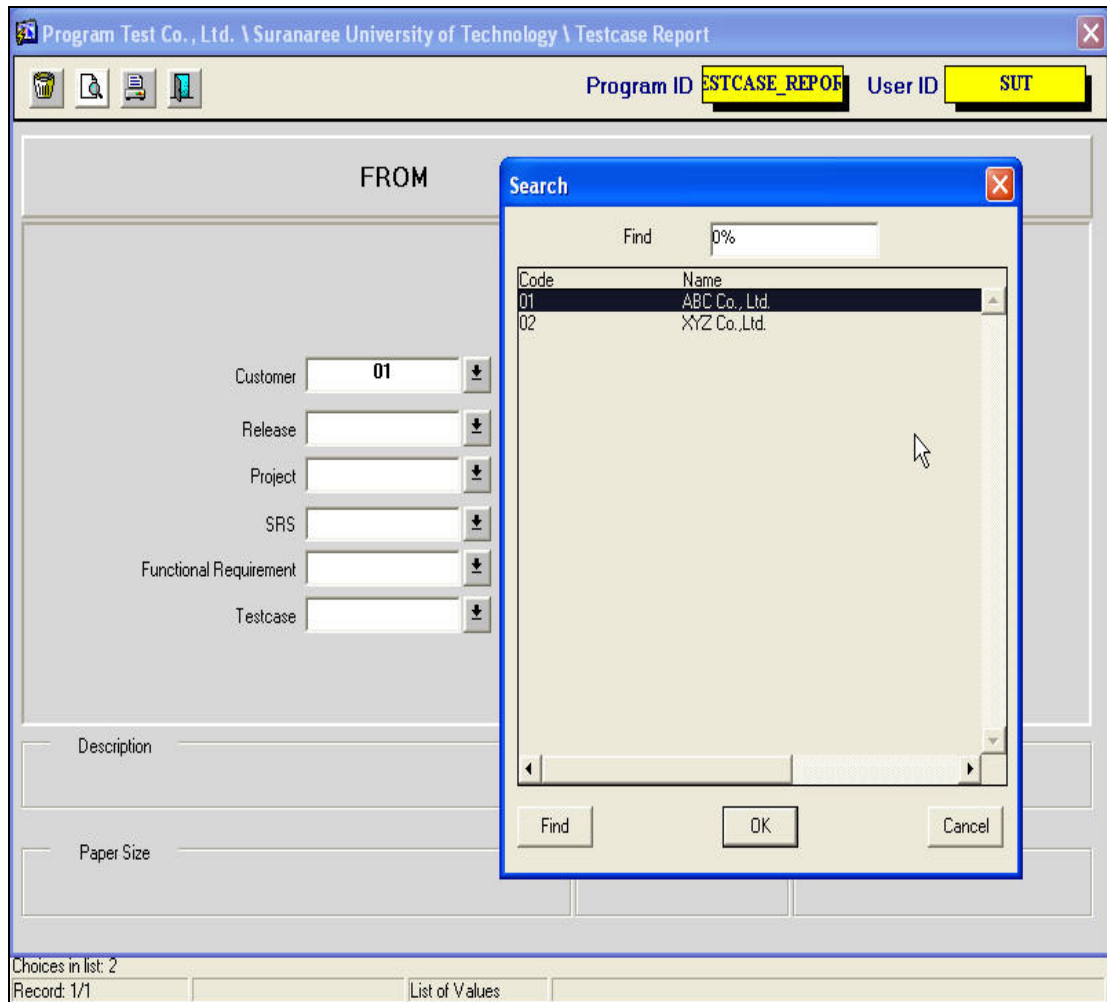
- 3) บันทึกในส่วนของกรณีทดสอบที่สัมพันธ์กับความต้องการด้านหน้าที่ โดยที่ผู้ทดสอบจะต้องทำการวิเคราะห์ความต้องการ และนำหลักการ การออกแบบ กรณีทดสอบแบบเบสิคบอกซ์ โดยใช้เทคนิคการสร้าง Thread Testing มาสร้างกรณีทดสอบ แล้วนำกรณีทดสอบที่ได้มาบันทึกลงในระบบ โดยกดปุ่มเงื่อนไข (Condition) โปรแกรมบันทึกกรณีทดสอบจะปรากฏ ดังแสดงในรูปที่ 4.7 ซึ่งข้อมูลที่ผู้ทดสอบบันทึกได้แก่ รหัสกรณีทดสอบ รายละเอียดของกรณีทดสอบ เงื่อนไขก่อนใช้กรณีทดสอบ เงื่อนไขหลังใช้กรณีทดสอบ รหัสข้อมูลเข้า รายละเอียดข้อมูลเข้า ผลลัพธ์ที่ต้องการ สถานะผ่าน/ไม่ผ่าน ข้อสมมุติฐานและข้อจำกัด และ การขึ้นตรงต่อกัน

The screenshot shows a 'Testcase' form with the following sections:

- Testcase No:** TC20060301
- Testcase Item:** Brief description of the test items exercised by this test case
- Pre-condition:** Conditions must be established before the test
- Post-condition:** Conditions must be established after the test
- Input No / Input Specification:**
 - INT01: Bullet items to describe each input required for this test case 01
 - INT02: Bullet items to describe each input required for this test case 02
- Expected output:** Bullet items to describe each expected output for this test case
- Pass/Fail Criteria:**
 - ☒ PASS
 - ☐ FAIL
- Assumption & Constraints:** Bullet items
- Dependencies:** List all the test cases depend on or are depend by this test case
- OK** button at the bottom.

รูปที่ 4.7 บันทึกในส่วนของกรณีทดสอบ

- 4) พิมพ์เอกสารกรณีทดสอบ เป็นส่วนที่ออกรายงานให้กับผู้ทดสอบ เพื่อนำไปใช้ในการทดสอบ ดังแสดงในรูปที่ 4.8



รูปที่ 4.8 พิมพ์เอกสารกรณีทดสอบ

Program ID : TESTCASE_REPORT	Program Test Co., Ltd.	Page : 1
	Testcase Report	Print Date : 09/05/2006 18:35
Customer : 01 ABC Co., Ltd.		
Release : 1 Release 1		
Project : 01 Project 1		
SRS : SRS20060301 Software Requirements Specification 01		
Functional Requirement : F20060301 Functional Requirements 01		
Testcase : TC20060301 Brief description of the test items exercised by this test case		
Pre-Condition : Conditions must be established before the test		
Post-Condition : Conditions must be established after the test		
Input No.	Input Specification	
INT01	Bullet items to describe each input required for this test case 01	
INT02	Bullet items to describe each input required for this test case 02	
Expected Output : Bullet items to describe each expected output for this test case		
Pass / Fail : PASS		
Assumption & Constraints : Bullet items		
Dependencies : List all the test cases depend on or are depend by this test case		

รูปที่ 4.9 เอกสารกรณีทดสอบ

4.6 อภิปรายผล

ในการทดสอบการสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ เครื่องมือสามารถสร้างกรณีทดสอบได้ และมีลำดับขั้นตอนในการทำงานที่ไม่ยุ่งยาก สามารถจัดเก็บรวบรวมข้อกำหนดความต้องการ และข้อมูลกรณีทดสอบไว้เป็นมาตรฐานเดียวกัน โดยเครื่องมือจะช่วยเก็บรวบรวมความต้องการที่มีความสัมพันธ์กรณีทดสอบ ทำให้ผู้ทดสอบมีความสะดวกรวดเร็วในการสรุปและประเมินผลการทดสอบ

บทที่ 5

บทสรุป

จากการศึกษา วิจัย และพัฒนาเครื่องมือซอฟต์แวร์ สามารถสรุปผลที่ได้รับ ปัญหา และข้อจำกัดที่พบได้

5.1 สรุปผลการวิจัย

ในการทดสอบโปรแกรมเพื่อให้สามารถค้นพบข้อผิดพลาดได้มากที่สุด ผู้ทดสอบต้องออกแบบและสร้างกรณีทดสอบให้ครอบคลุมทุกข้อผิดพลาดที่อาจเกิดขึ้นได้ ซึ่งเป็นงานที่ต้องใช้ความละเอียดรอบคอบ และยังใช้เวลานาน จากการวิจัยเครื่องมือนี้สามารถช่วยลดภาระในส่วนนี้ให้กับผู้ทดสอบได้ โดยเครื่องมือสามารถสร้างกรณีทดสอบตามลักษณะเฉพาะ หรือเงื่อนไขของข้อมูลได้ด้วยวิธีการของแบล็กบ็อกซ์

โดยในส่วนของผู้ทดสอบจะต้องวิเคราะห์และแยกชั้นสมมูลของข้อมูลเข้า ตามประเภทและลักษณะเฉพาะของข้อมูลเพื่อนำไปสร้างกรณีทดสอบ จากชั้นสมมูลที่ได้รับ ซึ่งจะช่วยให้สามารถวิเคราะห์ค่าขอบเขตของข้อมูล เพื่อให้ข้อมูลทดสอบที่สร้าง อยู่ในบริเวณที่อาจเกิดข้อผิดพลาดขึ้นได้มากที่สุด และสามารถสร้างกรณีทดสอบครอบคลุมทุกชั้นสมมูลที่ได้รับ

ส่วนฐานข้อมูลกรณีทดสอบ ซึ่งเก็บข้อมูลกรณีทดสอบ ข้อมูลโครงการทดสอบ ข้อมูลทดสอบ ผู้ทดสอบสามารถแก้ไข เพิ่ม หรือลดข้อมูลที่เก็บไว้ในฐานข้อมูลได้

ในขั้นตอนของการทดสอบแต่ละกรณีทดสอบ เครื่องมือจะช่วยออกรายงานให้แก่ผู้ทดสอบ ซึ่งจะสะดวกและรวดเร็วในการนำไปทำการทดสอบ

เครื่องมือนี้สามารถสรุปผลการทำงาน การทดสอบโปรแกรมสำหรับผู้ทดสอบแต่ละคน ทำให้ผู้จัดการโครงการ สามารถนำไปประเมินผลการทำงานของผู้ทดสอบแต่ละคนได้

นอกจากนี้ รูปแบบของข้อมูลที่ใช้ในเครื่องมือดังกล่าวเป็นรูปแบบที่ปรับปรุงมาจากมาตรฐาน IEEE Std.829-1998 ซึ่งเป็นมาตรฐานทำให้ทั้งผู้ทดสอบ และเจ้าของโครงการสามารถเข้าใจข้อมูลต่าง ๆ ได้ง่าย และเครื่องมือยังได้ออกแบบฐานข้อมูลไว้เชื่อมโยงกับเครื่องมือซอฟต์แวร์สำหรับจัดการกระบวนการวางแผนการทดสอบซอฟต์แวร์เพื่อให้การทดสอบนั้นมีประสิทธิภาพมากขึ้น

5.2 ประโยชน์ของเครื่องมือสร้างกรณิทดสอบสำหรับการทดสอบการยอมรับ

- (1) ช่วยเก็บบันทึก รวบรวมข้อมูลเกี่ยวกับกรณิทดสอบ ซึ่งสามารถนำมาใช้วัดประสิทธิภาพการทำงานของผู้ทดสอบ ติดตามผลการทดสอบ ความก้าวหน้าของการทดสอบ และคุณภาพของซอฟต์แวร์ที่พัฒนาขึ้นได้
- (2) มีการเก็บข้อมูลกรณิทดสอบที่เป็นมาตรฐานเดียวกันทำให้ผู้ทดสอบสามารถใช้งานหรือแก้ไขปรับปรุง กรณิทดสอบ ได้ง่ายขึ้น
- (3) กรณิทดสอบที่บันทึกไว้ สามารถนำไปใช้ในการทดสอบซอฟต์แวร์ที่พัฒนาขึ้นในอนาคตได้
- (4) สามารถนำเครื่องมือที่พัฒนาขึ้นไปใช้งานร่วมกับเครื่องมือที่ทำงานเกี่ยวข้องกันได้ เช่น เครื่องมือที่ช่วยวางแผนการทดสอบ

5.3 ข้อจำกัดของเครื่องมือที่สร้างขึ้น

- (1) ข้อมูลเข้าที่สามารถนำมาสร้างกรณิทดสอบได้นั้น ต้องมีลักษณะเฉพาะ และเงื่อนไขตามที่เครื่องมือกำหนดเท่านั้น
- (2) ข้อมูลเข้าที่นำมาสร้างกรณิทดสอบจะไม่พิจารณาถึงข้อมูลอื่นที่มีความสัมพันธ์กับข้อมูลเข้านั้น
- (3) กรณิทดสอบที่สร้างขึ้นมีจำนวนมาก ผู้ทดสอบอาจไม่สามารถนำไปทดสอบได้หมด ทำให้ความก้าวหน้าในการทดสอบ ของโครงการไม่มากนัก
- (4) ความสามารถของผู้ทดสอบ มีความจำเป็นเนื่องจากการต้องใช้ในการวิเคราะห์และแยกชั้นสมมูลของข้อมูลเข้า

5.4 แนวทางในการพัฒนาต่อ

เครื่องมือนี้สร้างกรณิทดสอบโดยใช้วิธีแบบแบล็กบ็อกซ์ซึ่งเป็นวิธีที่นิยมใช้ แต่ในบางกรณิข้อผิดพลาดที่เกิดขึ้นในโปรแกรมไม่สามารถตรวจสอบได้พบโดยกรณิทดสอบที่สร้างขึ้นด้วยวิธีนี้ เช่น การคำนวณตัวเลขผิดพลาด ซึ่งต้องใช้วิธีการทดสอบแบบไวท์บ็อกซ์ จึงสามารถพบข้อผิดพลาดนี้ การนำเครื่องมือนี้ไปพัฒนาต่อเพื่อให้สามารถสร้างกรณิทดสอบแบบไวท์บ็อกซ์ด้วย จะทำให้การทดสอบมีประสิทธิภาพดียิ่งขึ้น นอกจากนี้ ข้อมูลที่ใช้งานในระบบมีรูปแบบที่ปรับปรุงมาจาก มาตรฐานของ IEEE Std 829-1998 ซึ่งเป็นมาตรฐานของข้อมูลที่ใช้ในการทดสอบ ที่นำมาใช้กันอย่างกว้างขวาง สามารถนำไป พัฒนาต่อหรือนำไปรวมกับเครื่องมือซอฟต์แวร์ที่ช่วยในการทดสอบในส่วนอื่น ๆ ได้ง่าย เช่น เครื่องมือเพื่อจัดการการวางแผนการทดสอบซอฟต์แวร์

เพื่อให้การทดสอบนั้นมีประสิทธิภาพ รวดเร็ว และถูกต้องมากขึ้น อันจะทำให้ซอฟต์แวร์ที่ได้มี
คุณภาพดีขึ้นด้วย

รายการอ้างอิง

- น้ำฝน อัสวเมจิน (2542). รายงานการวิจัยฉบับสมบูรณ์: แนวทางสำหรับการเตรียมเอกสารข้อกำหนดความต้องการซอฟต์แวร์. กรุงเทพมหานคร: ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย.
- อภิรักษ์ จิรายุสกุล (2544). วิศวกรรมซอฟต์แวร์. กรุงเทพมหานคร : มหาวิทยาลัยรามคำแหง
- (1993). **Requirements Interdependencies and Stakeholders Preferences Resource.** [ออนไลน์]. ได้จาก: <http://www.inf.ethz.ch/personal/giesen/pub/re02.pdf>.
- (1996). **Meta Models For Requirements Engineering.** [ออนไลน์]. ได้จาก: <http://ksi.cpsc.ucalgary.ca/KAW/KAW96/jarke/Jarke.html>.
- (1997). **Linguistic Instruments for the Integration of Scenarios in Requirement Engineering.** [ออนไลน์]. ได้จาก: <http://citeseer.ist.psu.edu/update/362443>.
- (2003). **Clear Requirements Workbench.** [ออนไลน์]. ได้จาก: <http://www.livespecs.com/modules.php?op=modload&name=News&file=index&catid=4&topic=&allstories=1>.
- (2003). **Requirements Engineering - Concepts and Relations.** [ออนไลน์]. ได้จาก: http://www.cs.helsinki.fi/u/tevanlin/re_concepts.pdf.
- E.A. Giakoumakis and Xylomenos. (1996). **Evaluation and selection criteria for software requirements specification standards.** Software Engineering Journal, 307-319.
- Gerald Kotonya and Ian Sommerville. (2002). **Requirements Engineering Processes and Techniques.** John Wiley & Sons, Inc.
- Hiren D. Desai. **Test Case Management System (TCMS).** IEEE. vol.3, 28 Nov.-2 Dec. 1994: 1581 – 1585.
- Ian Sommerville. (1992). **Software Engineering.** Addison-Wesley Publishing Company, Fourth Edition.
- John Watkins. (2001). **Testing IT An off-the-shelf Software Testing Process.**
- Mercury Interactive Ltd. **TestDirector.** [ออนไลน์]. ได้จาก: <http://www.merc-int.com>.
- +1 Software Engineering. **+1 Test.** [ออนไลน์]. ได้จาก: <http://www.Plus-One.com>

ภาคผนวก ก

บทความผลงานวิจัยที่น่าสนใจในการประชุมวิชาการ
วิทยาศาสตร์และเทคโนโลยีแห่งประเทศไทย ครั้งที่ 32

การพัฒนาเครื่องมือสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ

DEVELOPMENT OF TEST CASE GENERATION TOOL FOR ACCEPTANCE TESTING

นริศ มิ่งโมรา, พิชโยทัย มัทธนาภีวัฒน์

School of Computer Engineering, Suranaree University of Technology, Nakhon Ratchasima

30000, Thailand.

E-mail:nrmimg@hotmail.com

บทคัดย่อ: การพัฒนาซอฟต์แวร์ ขั้นตอนหนึ่งที่สำคัญเพื่อให้ซอฟต์แวร์ที่พัฒนาขึ้นนั้น มีประสิทธิภาพน่าเชื่อถือ และตรงตามความต้องการของผู้ใช้งานคือ ขั้นตอนการทดสอบ ซึ่งขั้นตอนสุดท้ายของการทดสอบก่อนการส่งมอบงานอย่างเป็นทางการให้กับผู้ใช้งานคือ การทดสอบการยอมรับ สำหรับการทดสอบการยอมรับ สิ่งที่เป็นสำหรับผู้ทดสอบเพื่อใช้ในการทดสอบคือ กรณีทดสอบ ซึ่งการออกแบบกรณีทดสอบเป็นเรื่องที่ยุ่งยากและใช้เวลามากสำหรับผู้ทดสอบ ดังนั้นงานวิจัยนี้จึงมีแนวคิดในการออกแบบและพัฒนาเครื่องมือช่วยในการสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ โดยใช้วิธีการออกแบบกรณีทดสอบแบบแบล็กบ็อกซ์ ซึ่งผู้ทดสอบจะต้องทำขั้นตอนของการแยกชั้นสมมูล เพื่อแยกข้อมูลตามประเภท ซึ่งจะทำได้คลาสของข้อมูลเข้าที่ถูกต้องและไม่ถูกต้อง และทำการวิเคราะห์ค่าขอบเขตของข้อมูล เพื่อให้กรณีทดสอบที่สร้างขึ้นนั้นครอบคลุมค่าบริเวณขอบเขตสำหรับทุกคลาสของข้อมูล โดยในงานวิจัยนี้ได้นำวิธีการในการจัดเก็บความต้องการเชิงหน้าที่ ที่ได้มาจากข้อกำหนดความต้องการซอฟต์แวร์ซึ่งอ้างอิงมาจากมาตรฐานต่างประเทศ 4 มาตรฐาน ได้แก่ มาตรฐาน IEEE Std.830-1993 มาตรฐาน ESA PSS-05-0 มาตรฐาน NASA-DID-P200 และมาตรฐาน DOD-STD-498 มาสร้างเป็นกรณีทดสอบในส่วนของการทดสอบการยอมรับ ซึ่งเครื่องมือช่วยในการสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ จะช่วยลดภาระของผู้ทดสอบ และสามารถนำกรณีทดสอบกลับมาใช้ใหม่ได้.

Abstract: One of the important aspects of software development to make it reliable and conform to user requirement is testing. The last phase of testing is acceptance testing that uses black box technique to test the requirements specified in the software requirement specification. To generate test cases using black box testing testers have to perform Equivalence Partitioning to generate valid and invalid classes for each input data. Then testers have to do Boundary Value Analysis to ensure that the test cases cover all boundary values. The research also studied the methodology to store the functional requirement, IEEE Std.830-1993 standard, ESA PSS-05-0 standard, NASA-DID-P200 standard and DOD-STD-498 standard. The functional requirement from

the software requirement specification will be used to generate test cases for the acceptance testing. Actually, the design of test cases for the acceptance testing used considerable time to consider. This research designed and developed tool of test case generation for acceptance testing to help testers to test in this phase.

1. ความเป็นมา

ในกระบวนการพัฒนาซอฟต์แวร์จุดมุ่งหมายหลักคือ การที่กระบวนการพัฒนาซอฟต์แวร์สามารถผลิตผลิตภัณฑ์ซอฟต์แวร์ต่าง ๆ ได้อย่างถูกต้องและตรงตามความต้องการของลูกค้าหรือผู้ใช้งาน ดังนั้นในการพัฒนาซอฟต์แวร์จึงจำเป็นต้องมีกระบวนการสำคัญต่าง ๆ ไม่ว่าจะเป็นการค้นหาและรวบรวมความต้องการที่ถูกต้องจากผู้ใช้งาน การสร้างข้อกำหนดความต้องการซอฟต์แวร์ให้ตรงกับความความต้องการที่ได้รวบรวม การออกแบบและการเขียนโปรแกรมซอฟต์แวร์ให้สามารถทำงานได้ตรงตามข้อกำหนดความต้องการซอฟต์แวร์ที่ได้กำหนด และขั้นตอนหนึ่งที่สำคัญในการพัฒนาซอฟต์แวร์ เพื่อให้ซอฟต์แวร์ที่พัฒนาขึ้นนั้นมีประสิทธิภาพ น่าเชื่อถือ และถูกต้องตามความต้องการของผู้ใช้งาน คือ ขั้นตอนการทดสอบ (Testing) การทดสอบซอฟต์แวร์ซึ่งมีได้หลายระดับ ตั้งแต่ การทดสอบระดับหน่วย (Unit testing) การทดสอบการรวม (Integration testing) การทดสอบระบบ (System Test) การทดสอบการยอมรับ (Acceptance Testing) และการทดสอบย้อนกลับ (Regression testing) ในการทดสอบทุกระดับสิ่งที่จำเป็นสำหรับผู้ทดสอบ (Tester) ใช้ในการทดสอบคือ กรณีทดสอบ (Test Case)

กรณีทดสอบที่ดีนั้นจะต้องสามารถค้นหาข้อผิดพลาดที่ยังไม่ถูกค้นพบในซอฟต์แวร์ที่พัฒนานั้นได้ การออกแบบกรณีทดสอบในแต่ละระดับแบ่งออกเป็นสองประเภท คือ

- แบบ แบล็กบ็อกซ์ (Black-box testing)
- แบบ ไวท์บ็อกซ์ (White-box testing)

ซึ่งการทดสอบการยอมรับ (Acceptance Testing) จะสัมพันธ์กับช่วงของ Requirements Phase ของ V Model ซึ่ง Phase ของการทำ Requirements นั้นจะต้องมีการวางแผน การทบทวนเพื่อตรวจสอบ Requirement และกำหนดตัวแทนของ User ไว้ล่วงหน้าก่อนทำการทดสอบการยอมรับ (Acceptance Testing)

งานวิจัยนี้จึงมีแนวคิดในการออกแบบและพัฒนาเครื่องมือช่วยในการสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ โดยในงานวิจัยนี้จะศึกษาถึงวิธีการในการจัดเก็บความต้องการเชิงหน้าที่ (Functional Requirements) และนำมาสร้างกรณีทดสอบในส่วนของการทดสอบการยอมรับ เนื่องจากข้อกำหนดความต้องการซอฟต์แวร์นั้นเป็นพื้นฐานในการออกแบบ การเขียนโปรแกรม และการทดสอบต่าง ๆ ดังนั้นจะเห็นได้ว่าข้อกำหนดความต้องการซอฟต์แวร์เป็นปัจจัยที่สำคัญต่อความสำเร็จในการพัฒนาซอฟต์แวร์เป็นอย่างมาก อย่างไรก็ตามงานวิจัยนี้จะศึกษาในส่วนของความต้องการเชิงหน้าที่ ที่มีผลต่อการทดสอบเพื่อการยอมรับเท่านั้น.

2. แนวคิดเกี่ยวกับงานและขั้นตอนการสร้างเครื่องมือสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ

เนื่องจากการวิเคราะห์ระบบงานใดระบบงานหนึ่งนั้น การที่จะให้งานออกมาสมบูรณ์ และถูกต้อง ผู้พัฒนาซอฟต์แวร์จำเป็นต้องมีความรู้ ความสามารถ และประสบการณ์ต่าง ๆ ในการวิเคราะห์ระบบงานที่ดี เพื่อส่งมอบงานให้นักออกแบบระบบทำการออกแบบระบบต่อไป ผู้พัฒนาซอฟต์แวร์ต้องทำการสอบถามความต้องการจากผู้ใช้งาน ว่าต้องการให้ซอฟต์แวร์มีหน้าที่การทำงานเป็นอย่างไร ซึ่งผู้พัฒนาซอฟต์แวร์จะจดบันทึกความต้องการและทำการวิเคราะห์ระบบ โดยผู้พัฒนาซอฟต์แวร์จะต้องเขียนเอกสารข้อกำหนดความต้องการซอฟต์แวร์ (Software Requirements Specification Document) ซึ่งเอกสารข้อกำหนดความต้องการซอฟต์แวร์ คือเอกสารที่ถูกพัฒนาขึ้นมาหลังจากทราบความต้องการของผู้ใช้งานแล้ว ซึ่งผู้พัฒนาซอฟต์แวร์เขียนรายละเอียดต่าง ๆ เกี่ยวกับซอฟต์แวร์ในเอกสารนี้ เพื่ออธิบายถึงหน้าที่และการทำงานของซอฟต์แวร์ โดยจะต้องครอบคลุมถึงความต้องการของผู้ใช้งานด้วย

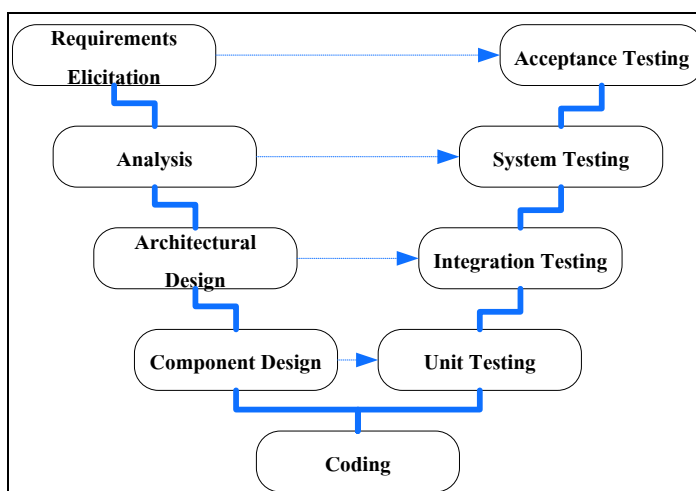
ความต้องการซอฟต์แวร์ (Software Requirements) คือคุณลักษณะหรือความสามารถของซอฟต์แวร์ที่ถูกกำหนด เพื่อใช้สำหรับแก้ปัญหาหรือดำเนินงานให้บรรลุเป้าหมายตามความต้องการของผู้ใช้ โดยที่ความสามารถเหล่านั้นจะถูกกำหนดในรูปของสัญญา ข้อกำหนด มาตรฐาน หรือเอกสารมาตรฐานอื่นที่กำหนด โดยที่ความต้องการนั้นสามารถแบ่งออกเป็น 2 ประการคือ

- ความต้องการเชิงหน้าที่ (Functional Requirements)
คือ การทำเอกสารใน ข้อกำหนดความต้องการซอฟต์แวร์ (Software Requirements Specification - SRS) ซึ่งจะบรรยายถึงความเป็นไปได้ของความคาดหวังเกี่ยวกับพฤติกรรมภายนอกของระบบซอฟต์แวร์ ซึ่งจะถูกใช้ในการพัฒนา, ตรวจสอบ, ประกันคุณภาพ, การจัดการโครงการ และความสัมพันธ์ของฟังก์ชันในโครงการ
- ความต้องการที่ไม่ใช่ความต้องการเชิงหน้าที่ (Non-Functional Requirements) คือ สิ่งที่กำหนดถึงประสิทธิภาพของซอฟต์แวร์ ข้อบังคับ และมาตรฐานที่ใช้ในระหว่างการพัฒนาซอฟต์แวร์

การทดสอบการยอมรับ (Acceptance Testing) วัตถุประสงค์ของการทดสอบการยอมรับ เพื่อให้ยืนยันว่าซอฟต์แวร์ ที่ทดสอบนั้นสามารถทำงานได้ตรงตามความต้องการของผู้ใช้งาน และสร้างความเชื่อมั่นว่าระบบสามารถทำงานได้ถูกต้องก่อนที่จะส่งมอบอย่างเป็นทางการให้แก่ผู้ใช้งาน. การทดสอบการยอมรับ ถูกดำเนินโดยตัวแทนของผู้ใช้งาน อาจจะเป็นคนเดียวหรือมากกว่า ก็ได้ ซึ่งจะได้รับความช่วยเหลือจากทีมงานของผู้ทดสอบ. การทดสอบการยอมรับ ใช้เทคนิคแบบ แบล็กบ็อกซ์ (Black-box testing) โดยใช้เทคนิคการสร้าง Thread Testing เป็นลักษณะของการทดสอบตามงานที่ทำ ในลักษณะของการใช้งานระบบ เพื่อตรวจสอบความต้องการทางด้าน

ธุรกิจในระดับสูง (high-level business requirement). นอกจากนี้ การทดสอบการยอมรับจะต้องดูจุดที่ต้องทดสอบตามเอกสารของระบบ โดยตัวแทนผู้ใช้งาน. การทดสอบการยอมรับจะสัมพันธ์กับช่วงของขั้นตอนการเก็บรวบรวมความต้องการของ V-Model

V-Model เป็นกระบวนการพัฒนา และทดสอบซอฟต์แวร์ และที่สำคัญคือ การวางแผน และเตรียมการทดสอบ ตั้งแต่ขั้นตอนการพัฒนา ซึ่งในฝั่งซ้ายของ V-Model จะเป็นขั้นตอนการพัฒนาซอฟต์แวร์แบบน้ำตก (Waterfall Model) และทางด้านขวาจะเป็นขั้นตอนของการทดสอบ



รูปที่ 1.1 V-Model

ความสัมพันธ์ของการทดสอบการยอมรับกับความต้องการ

การทดสอบการยอมรับจะสัมพันธ์ กับช่วงของ Requirements Phase ของ V-Model ซึ่ง Phase ของการทำ Requirements Elicitation นั้นจะต้องมีการวางแผน การทบทวน เพื่อตรวจสอบความต้องการ และกำหนดตัวแทนของผู้ใช้งานไว้ล่วงหน้าก่อนทำการทดสอบการยอมรับ

กรณีทดสอบ (Test case) คือข้อมูลที่ผู้ทดสอบใช้ทำการทดสอบรายการ (Item) ใด ๆ เพื่อให้ทราบว่าการทำงานของรายการนั้นถูกต้องหรือไม่ ประกอบด้วยข้อมูลเข้า (Input) ผลที่คาดว่าจะได้ (Expected output) และข้อจำกัดต่าง ๆ ในการใช้กรณีทดสอบนั้น กรณีทดสอบที่ดีต้องมีคุณสมบัติคือ มีความน่าจะเป็นสูงที่จะเจอข้อผิดพลาด (Error) ซึ่งยังไม่ถูกค้นพบในโปรแกรม และกรณีทดสอบที่ประสบความสำเร็จ คือกรณีทดสอบที่สามารถค้นพบข้อผิดพลาดที่เกิดขึ้นในโปรแกรมได้

การเลือกกรณีทดสอบเพื่อนำไปทำการทดสอบ มีหลักการในการพิจารณาดังนี้ (3)

1) กรณีทดสอบที่ใช้ทดสอบความสามารถ ความมั่นคงของระบบมีความสำคัญมากกว่ากรณีทดสอบที่ใช้ทดสอบส่วนประกอบของระบบ เนื่องจากผู้ใช้จะให้ความสำคัญกับความมั่นคงของระบบในการทำงานมากกว่า ข้อผิดพลาดเล็กน้อยที่เกิดขึ้นในระบบ ดังเช่นข้อผิดพลาดที่ทำให้เกิดการสูญเสีย หรือสูญหาย ของข้อมูล มีความสำคัญมากกว่าข้อผิดพลาดที่ทำให้ข้อมูลที่แสดงทางจอภาพขาดหายไป

2) ในกรณีที่มีการปรับปรุง หรือแก้ไขระบบใหม่ กรณีทดสอบที่ใช้ทดสอบความสามารถเดิมของระบบ มีความสำคัญมากกว่ากรณีทดสอบสำหรับใช้ทดสอบความสามารถของระบบที่พัฒนาขึ้นใหม่

3) ในกรณีที่มิมีข้อจำกัดในเรื่องของจำนวนกรณีทดสอบ กรณีทดสอบที่มีข้อมูลทดสอบในลักษณะทั่วไปมีความสำคัญมากกว่ากรณีทดสอบที่มีข้อมูลทดสอบอยู่ในช่วงที่เป็นค่าขอบเขต

เทคนิคที่ใช้ในการออกแบบกรณีทดสอบ (4)

การออกแบบกรณีทดสอบในแต่ละระดับแบ่งออกเป็นสองประเภท คือ

- การออกแบบกรณีทดสอบแบบไวยากรณ์ เป็นการสร้างกรณีทดสอบโดยพิจารณาจาก โครงสร้าง (Structure) ของโปรแกรม โดยสร้างกรณีทดสอบที่ครอบคลุมทุกเส้นทาง (Path) ตามโครงสร้างของโปรแกรม เพื่อให้แน่ใจว่าทุกคำสั่งในโปรแกรมได้รับการทดสอบอย่างน้อยหนึ่งครั้ง และทุกคำสั่งที่เป็นเงื่อนไขถูกทดสอบ ทั้งกรณีที่เงื่อนไขนั้นเป็นจริงและกรณีที่เงื่อนไขนั้นเป็นเท็จ นอกจากนี้ยังสามารถค้นพบข้อผิดพลาดที่เกิดขึ้นจากข้อมูลภายในโมดูลได้อีกด้วย การสร้างกรณีทดสอบด้วยวิธีการของไวยากรณ์ ผู้ทดสอบต้องสร้างกราฟการควบคุมกระแส (Control flow graph) จากโปรแกรมที่พัฒนาขึ้น และนำกราฟที่ได้มาสร้างกรณีทดสอบเพื่อให้อุปกรณ์ครอบคลุมทุกคำสั่ง ทุกเส้นทาง และทุกเงื่อนไขของโปรแกรม
- การออกแบบกรณีทดสอบแบบแบล็กบ็อกซ์ เป็นการสร้างกรณีทดสอบโดยพิจารณาจากข้อกำหนด (Specification) และเงื่อนไข (Condition) ของข้อมูลนั้น การทดสอบสนใจเฉพาะข้อมูลเข้าและผลลัพธ์ของโมดูลที่ต้องการทดสอบ เพื่อให้ตรงตามข้อกำหนดที่ได้ออกแบบไว้เท่านั้น ซึ่งจะไม่พิจารณาถึงโครงสร้างภายในโมดูลนั้น

ในการศึกษาและการทำวิจัยเกี่ยวกับข้อกำหนดความต้องการซอฟต์แวร์ ได้มีผู้ทำการศึกษาเกี่ยวกับเรื่องนี้เป็นจำนวนมาก ดังจะเห็นได้จาก มีบทความเกี่ยวกับการศึกษาวิจัยในเรื่องนี้ในจุลสารต่าง ๆ ทั้งในและต่างประเทศ สำหรับการศึกษาและวิจัยในครั้งนี้ ผู้วิจัยได้ศึกษาแนวทางสำหรับการเตรียมเอกสารข้อกำหนดความต้องการซอฟต์แวร์ (2), ระบบจัดการกรณีทดสอบ (Hiren

D. Desai., Test Case Management System (TCMS), 2537) และ Testing IT An off-the-shelf Software Testing Process (4) รวมทั้งทำการศึกษาการใช้งาน โปรแกรม Test Director 6 (Mercury Interactive Ltd., [Http://www.merc-int.com](http://www.merc-int.com))

สำหรับขั้นตอนในการออกแบบเครื่องมือสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ มีขั้นตอนดังนี้

1. ทำการศึกษาแนวทางสำหรับการเตรียมเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ซึ่งอ้างอิงมาตรฐานจากต่างประเทศทั้งหมด 4 มาตรฐาน ได้แก่ มาตรฐาน IEEE Std.830-1993 มาตรฐาน ESA PSS-05-0 มาตรฐาน NASA-DID-P200 และมาตรฐาน DOD-STD-498 (1) ได้สรุปผลการประเมินมาตรฐานทั้ง 4 มาตรฐานดังตารางที่ 1.1

ตารางที่ 1.1 แสดงผลการประเมินมาตรฐานทั้ง 4 มาตรฐาน

เกณฑ์	การประเมินผล			
	IEEE Std.830-1993	ESA PSS-05-0	NASA-DID-P200	DOD-STD-498
ความเป็นอิสระ	ปานกลาง	ปานกลาง	สูง	สูง
การรวมเป็นหนึ่งเดียว	ปานกลาง	ปานกลาง	สูง	สูง
ความถูกต้อง	สูง	ต่ำ	สูง	สูง
คุณลักษณะโดยทั่วไป	ไม่เหมาะสมกับโครงการขนาดใหญ่	ไม่เหมาะสมกับโครงการขนาดใหญ่	ไม่เหมาะสมกับโครงการขนาดเล็ก	ไม่เหมาะสมกับโครงการขนาดเล็ก
การจัดระเบียบของเอกสาร	สูง	ต่ำ	ต่ำ	ปานกลางถึงสูง
ความสมบูรณ์ของเนื้อหา	สูง	ปานกลาง	สูง	สูง
ความสมบูรณ์ของมุมมองภายนอก	เหมาะกับผู้ใช้ทุกคน	เหมาะกับนักทดสอบระบบเท่านั้น	เหมาะกับนักออกแบบระบบและนักทดสอบระบบ	เหมาะกับผู้ใช้ทุกคน
การเปลี่ยนแปลงแก้ไขได้	เพียงพอ	ไม่เพียงพอ	ไม่เพียงพอ	เพียงพอ

พร้อมทั้งศึกษางานวิจัยเกี่ยวกับการพัฒนาแนวทางสำหรับการเตรียมเอกสารข้อกำหนดความต้องการซอฟต์แวร์ (2) ซึ่งสรุปผลสำหรับเนื้อหาที่ควรมีในเอกสารข้อกำหนดความต้องการซอฟต์แวร์จำนวน 46 เนื้อหา โดยแบ่งเป็น 4 ส่วน อันได้แก่

- บทนำ ประกอบไปด้วย จุดประสงค์ ขอบเขต คำจำกัดความ รหัสพจน์ คำย่อ เอกสารอ้างอิง และภาพรวม
- รายละเอียดโดยรวม ประกอบไปด้วย การต่อประสานกับผู้ใช้ การต่อประสานกับฮาร์ดแวร์ การต่อประสานกับซอฟต์แวร์ การต่อประสานกับระบบสื่อสาร ข้อจำกัดของหน่วยความจำ การดำเนินงาน ความต้องการด้านการติดตั้งซอฟต์แวร์

หน้าที่ของซอฟต์แวร์ ข้อกำหนดในการเตรียมกระบวนการในการพิจารณาซอฟต์แวร์ ความสามารถในการติดตามความต้องการ ข้อกำหนด ข้อเสนอแนะและการขึ้นต่อกัน และการแบ่งส่วนปันส่วนความต้องการ

- ความต้องการที่เฉพาะเจาะจง ประกอบไปด้วย การต่อประสานภายนอก ความต้องการด้านหน้าที่ ความต้องการด้านประสิทธิภาพ ความต้องการด้านฐานข้อมูลเชิงตรรกะ ความต้องการด้านการดำเนินการ ความต้องการด้านการทวนสอบ ความต้องการด้านการยอมรับ ความต้องการด้านเอกสาร ความต้องการด้านความปลอดภัย ความต้องการด้านการใช้ทรัพยากรฮาร์ดแวร์คอมพิวเตอร์ให้เกิดประโยชน์สูงสุด ความต้องการด้านระบบสื่อสารคอมพิวเตอร์ ปัจจัยที่มีผลต่อคุณภาพของซอฟต์แวร์ ความต้องการด้านการฝึกอบรมบุคคลที่เกี่ยวข้อง ความต้องการด้านการเตรียมขั้นตอนต่าง ๆ ที่ใช้ในการพัฒนาซอฟต์แวร์ ความต้องการด้านบรรจุกัมพธ์ซอฟต์แวร์ ข้อกำหนดในการใช้งานซอฟต์แวร์ มาตรฐานที่มีในองค์กร ความเชื่อถือได้ สภาพพร้อมใช้งาน ความมั่นคง ความสามารถในการบำรุงรักษา และ ความสามารถในการใช้ได้หลายระบบ
 - ข้อมูลสนับสนุน ประกอบไปด้วย สารบัญ บทนิยาม ภาคผนวก และดัชนี
2. ในการพัฒนาเครื่องมือสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ ผู้วิจัยได้ทำการออกแบบส่วนต่าง ๆ ของเครื่องมือ ออกเป็น 3 ส่วน ได้แก่ ทำการออกแบบฐานข้อมูลที่ใช้เก็บกรณีทดสอบ (Database Design) พัฒนาตัวรับข้อมูลกรณีทดสอบ (Input Design) และตัวสร้างกรณีทดสอบ (Output Design)
- ในส่วนของการออกแบบ ผู้วิจัย ได้นำเอาข้อมูลที่ได้จาก เอกสารข้อกำหนดความต้องการซอฟต์แวร์ มาทำการออกแบบฐานข้อมูล
 - ในส่วนของการพัฒนาส่วนรับข้อมูลกรณีทดสอบ ผู้วิจัย ได้ทำการออกแบบโปรแกรมในการรับข้อมูลไว้ดังรูปที่ 1.2

รูปที่ 1.2 โปรแกรมบันทึกข้อมูลเอกสารข้อกำหนดความต้องการซอฟต์แวร์

ซึ่งโปรแกรมนี้จะทำการรับข้อมูลจากเอกสารข้อกำหนดความต้องการซอฟต์แวร์ หลังจากนั้นจะมีการระบุถึง กรณีทดสอบตามความต้องการเชิงหน้าที่ที่ได้ระบุไว้ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ซึ่งโปรแกรมในส่วนของการบันทึกกรณีทดสอบจะมีลักษณะดังรูปที่ 1.3

รูปที่ 1.3 โปรแกรมบันทึกข้อมูลเอกสารข้อกำหนดความต้องการซอฟต์แวร์

- ในส่วนของตัวสร้างกรณีทดสอบนั้น ระบบจะนำข้อมูลที่ได้จากการบันทึกมาทำการออกเป็นเอกสาร ดังแสดงในรูปที่ 1.4

Program ID : TESTCASE_REPORT	Program Test Co., Ltd. Testcase Report	Page : 1 Print Date : 09/05/2006 18:35
Customer : 01 ABC Co., Ltd.		
Release : 1 Release 1		
Project : 01 Project 1		
SRS : SRS20060301 Software Requirements Specification 01		
Functional Requirement : F20060301 Functional Requirements 01		
Testcase : TC20060301 Brief description of the test items exercised by this test case		
Pre-Condition : Conditions must be established before the test		
Post-Condition : Conditions must be established after the test		
Input No.	Input Specification	
INT01	Bullet items to describe each input required for this test case 01	
INT02	Bullet items to describe each input required for this test case 02	
Expected Output : Bullet items to describe each expected output for this test case		
Pass / Fail : PASS		
Assumption & Constraints : Bullet items		
Dependencies : List all the test cases depend on or are depend by this test case		

รูปที่ 1.4 แบบแสดงกรณีทดสอบ

3. สรุปผล

ในการทดสอบโปรแกรมเพื่อให้สามารถค้นพบข้อผิดพลาดได้มากที่สุด ผู้ทดสอบต้องออกแบบและสร้างกรณีทดสอบให้ครอบคลุมทุกข้อผิดพลาดที่อาจเกิดขึ้นได้ ซึ่งเป็นงานที่ต้องใช้ความละเอียด รอบคอบ และยังใช้เวลามาก จากการวิจัยเครื่องมือนี้สามารถช่วยลดภาระในส่วนนี้ให้กับผู้ทดสอบได้ โดยเครื่องมือสามารถสร้างกรณีทดสอบตามลักษณะเฉพาะ หรือเงื่อนไขของข้อมูลได้ ด้วยวิธีการของแบล็กบ็อกซ์

ในส่วนตัวสร้างกรณีทดสอบ สามารถสร้างข้อมูลทดสอบ จากข้อมูลที่ได้รับ และสามารถสร้างกรณีทดสอบครอบคลุมทุกเงื่อนไขที่ได้รับ

ส่วนฐานข้อมูลกรณีทดสอบ ซึ่งเก็บข้อมูลกรณีทดสอบ สามารถแก้ไข เพิ่ม หรือลดข้อมูลที่เก็บไว้ในฐานข้อมูลได้

ประโยชน์ของเครื่องมือระบบจัดการกรณีทดสอบซอฟต์แวร์

1) ช่วยลดภาระผู้ทดสอบ เนื่องจากระบบสามารถสร้างกรณีทดสอบที่ครอบคลุมทุกเงื่อนไข สำหรับวิธีการแบบแบล็กบ็อกซ์ได้

2) ช่วยเก็บบันทึก รวบรวมข้อมูลเกี่ยวกับกรณีทดสอบ ซึ่งสามารถนำมาใช้วัดประสิทธิภาพการทำงานของผู้ทดสอบ และคุณภาพของซอฟต์แวร์ที่พัฒนาขึ้นได้

3) มีการเก็บข้อมูลกรณีทดสอบที่เป็นมาตรฐานเดียวกันทำให้ผู้ทดสอบสามารถใช้งานหรือแก้ไขปรับปรุง กรณีทดสอบ ได้ง่ายขึ้น

4) กรณีทดสอบที่บันทึกไว้สามารถนำไปใช้ในการทดสอบซอฟต์แวร์ที่พัฒนาขึ้นในอนาคตได้

5) สามารถนำเครื่องมือที่พัฒนาขึ้นไปใช้งานร่วมกับเครื่องมือที่ทำงานเกี่ยวข้องกันได้ เช่น เครื่องมือที่ช่วยวางแผนการทดสอบ

นอกจากนี้ ในการวิจัยครั้งนี้ ผู้วิจัยได้นำแนวทางสำหรับการเตรียมเอกสารข้อกำหนดความต้องการซอฟต์แวร์มาใช้ เพื่อให้ผู้พัฒนาซอฟต์แวร์จัดทำเอกสารข้อกำหนดความต้องการซอฟต์แวร์ให้เข้ากับลักษณะการทำงานของผู้พัฒนาซอฟต์แวร์ในประเทศอีกด้วย.

เอกสารอ้างอิง

- (1) Giakoumakis, E.A. and Xylomenos (1996) "Evaluation and selection criteria for software requirements specification standards", Software Engineering Journal, P307-319.
- (2) น้ำฝน อัสวเมณิน. (1999), "แนวทางสำหรับการเตรียมเอกสารข้อกำหนดความต้องการซอฟต์แวร์" ลิขสิทธิ์ของบัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย.
- (3) Ian Sommerville. (1992) Software Engineering, Addison-Wesley Publishing Company, Fourth Edition.
- (4) John Watkins. (2001) Testing IT An off-the-shelf Software Testing Process.

ภาคผนวก ข

ตัวอย่างกรณีทดสอบ ของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติ

ในส่วนนี้ แสดงถึงตัวอย่างของข้อกำหนดความต้องการซอฟต์แวร์และกรณีทดสอบของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติ ซึ่งข้อมูลที่ได้จะนำไปบันทึกลงในโปรแกรมบันทึกข้อกำหนดความต้องการซอฟต์แวร์

ระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติ

ส่วนที่ 1

1. จุดประสงค์

SRS ฉบับนี้ อธิบายถึงความต้องการด้านหน้าที่และความต้องการด้านประสิทธิภาพในระบบย่อยของระบบสั่งซื้อสินค้า

2. ขอบเขต

ข้อกำหนดฉบับนี้ ใช้เป็นพื้นฐานในการออกแบบซอฟต์แวร์ ซึ่งสามารถสนับสนุนการทำงานของเครื่องคอมพิวเตอร์ได้ถึง 100 เครื่อง โดยผู้ใช้ได้รับการฝึกอบรมในการใช้เครื่องคอมพิวเตอร์เหล่านี้

3. คำจำกัดความ

- ลูกค้า = เป็นผู้ที่ต้องการซอฟต์แวร์ที่ได้รับการพัฒนาแล้ว
- ผู้ใช้ = เป็นผู้ใช้ประโยชน์จากซอฟต์แวร์ที่สมบูรณ์แล้ว ซึ่งอาจเป็นคนเดียวกับลูกค้าก็ได้

4. รหัสพจน์

- IEEE = Institute of Electrical and Electronics Engineers
- SDS = Software Design Specification
- SRS = Software Requirements Specification

5. คำย่อ

- CID = customer identification number
- PID = product identification number
- Std = Standard

6. เอกสารอ้างอิง

- IEEE Std 610.12-1990, IEEE Standard Glossary for Software Engineering Terminology (ANSI).
- IEEE Std-830, IEEE New York, 1984 and 1993.

7. ภาพรวม

ระบบย่อยของระบบสั่งซื้อสินค้า เป็นส่วนประกอบหนึ่งของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติ คำอธิบายของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติอยู่ในเอกสารที่มีชื่อเรื่องว่า "คำอธิบายการออกแบบของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติ ครั้งที่ 4 ตุลาคม 1996" ความต้องการที่ระบุในเอกสารฉบับนี้ได้รับการยอมรับเมื่อวันที่ 12 กันยายน 1995 และในส่วนที่ 3.8 ของข้อกำหนดนี้สามารถตรวจสอบความต้องการ หรือมีการทดสอบความต้องการกลับไปยังข้อกำหนดระบบของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติของบริษัท

ส่วนที่ 2

8. การต่อประสานกับผู้ใช้

การต่อประสานไปยังระบบย่อยของระบบสั่งซื้อสินค้า จะติดต่อโดยใช้คีย์บอร์ด ซึ่งคีย์บอร์ดนี้ มีฟังก์ชันพิเศษเพื่อลดจำนวนครั้งของการกดแป้นคีย์บอร์ด ซึ่งจะส่งผลกระทบต่อ การป้อนข้อมูลรายการเปลี่ยนแปลงของระบบสั่งซื้อสินค้า มาตรฐานของคีย์บอร์ด ใช้มาตรฐาน QWERTY สำหรับรายละเอียดต่าง ๆ อ้างอิงไปยังแผนผังระบบของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติ

9. การต่อประสานกับฮาร์ดแวร์

ระบบย่อยของระบบสั่งซื้อสินค้าจะพิมพ์รายงานออกทางเครื่องพิมพ์ ดังนั้น เครื่องคอมพิวเตอร์และเครื่องพิมพ์ต้องมีการต่อประสานทางด้านฮาร์ดแวร์อย่างง่าย

10. การต่อประสานกับซอฟต์แวร์

ระบบย่อยของระบบสั่งซื้อจะต่อประสานกับระบบย่อย ดังต่อไปนี้

- คลังสินค้า
- การบัญชี
- การตลาดและการขาย
- สมุดแจ้งรายการสินค้า
- ควบคุมเครือข่าย

11. การต่อประสานกับระบบสื่อสาร

การสื่อสารระหว่างเครื่องคอมพิวเตอร์กับอุปกรณ์ต่าง ๆ ใช้โปรโตคอล IEEE-488

12. ข้อกำหนดของหน่วยความจำ

ซอฟต์แวร์นี้ต้องการเนื้อที่ในฮาร์ดดิสก์ไม่ต่ำกว่า 1 กิกะไบต์ และต้องการหน่วยความจำมากกว่า 32 เมกกะไบต์ขึ้นไป

13. การดำเนินการ

- เครื่องคอมพิวเตอร์ที่ใช้ในระบบสั่งซื้อสินค้าจะเปิดแบบเชื่อมต่อตรง เป็นเวลา 16 ชั่วโมง ต่อ 1 วัน
- สัญญาณเตือนจะส่งเสียงดังที่เครื่องคอมพิวเตอร์ เมื่อมีเหตุการณ์ผิดปกติเกิดขึ้น จากนั้นจะทำการสำรองข้อมูล เพื่อนำข้อมูลไปใช้ในภายหลังได้
- เพิ่มข้อมูลของเหตุการณ์ปกติ ใช้เป็นต้นแบบในการสร้างมาตรฐานของรายงานและสามารถเข้าถึงข้อมูลโดยผู้ที่มีสิทธิ์เท่านั้น ระเบียบต่าง ๆ ของเหตุการณ์ปกติ ควรจะแสดงบนหน้าจอคอมพิวเตอร์หรือพิมพ์ออกทางเครื่องพิมพ์

14. ความต้องการด้านการติดตั้งซอฟต์แวร์

ไม่สามารถใช้ให้เป็นประโยชน์ได้ (Not application :N/A)

15. หน้าที่ของซอฟต์แวร์

ในระบบสั่งซื้อสินค้า ผู้ดำเนินการหรือเสมียนต้องทำงานหรือดำเนินงานตามขั้นตอนดังต่อไปนี้ หลังจากที่ได้ฝ่ายขายได้ทำการตกลงรายละเอียดการขายกับทางลูกค้าเรียบร้อยแล้ว ทางผู้ดำเนินการหรือเสมียนจะนำรายการดังกล่าวมาทำการบันทึกใบสั่งขายเก็บไว้ โดยระบบจะทำการตรวจสอบข้อมูลต่าง ๆ เพื่อการควบคุมไว้ 3 ส่วนคือ

- การตรวจสอบสถานะลูกค้า โดยแยกออกเป็น 2 ระดับ คือ Blacklist - ระบบจะไม่อนุญาตให้ทำการบันทึกใบสั่งขายเลย, Lock - ระบบจะยอมให้บันทึกใบสั่งขายนั้น ๆ แต่จะไม่สามารถออกใบจัดสินค้าได้จนกว่าทางแผนกบัญชีจะทำการอนุมัติรายการสั่งขายนั้น ๆ ก่อน
- การตรวจสอบวงเงินเครดิต เนื่องจากการเปิดลูกค้าใหม่จะต้องมีข้อตกลงระหว่างแผนกขายและแผนกบัญชีว่า ลูกค้าแต่ละรายจะอนุมัติวงเงินเครดิตให้เท่าไร และกำหนดไว้ในแฟ้มลูกค้า ระบบจะนำยอดหนี้ค้างชำระที่ค้างอยู่ บวกกับใบสั่งขายที่ยังไม่ออกอินวอยส์มาตรวจสอบกับวงเงินเครดิต ถ้าเกินใบสั่งขายดังกล่าวจะต้องผ่านการอนุมัติจากทางแผนกบัญชีก่อนเสมอ
- การตรวจสอบหนี้เกินกำหนดชำระ ระบบจะให้กำหนดมาตรฐานว่านโยบายของทางบริษัทจะอนุญาตให้ลูกค้าค้างชำระหนี้เกินจากเครดิตเทอมได้กี่วัน ดังนั้นถ้าลูกค้าในใบสั่งซื้อนี้มีการค้างชำระเกินจากที่กำหนด ใบสั่งขายดังกล่าวนี้จะต้องไปผ่านการอนุมัติจากทางแผนกบัญชีก่อนด้วยเช่นกัน
- หลังจากใบสั่งขายสมบูรณ์แล้วทางแผนกขายจะทำการบันทึกใบจัดสินค้าซึ่งระบบจะทำการตรวจสอบปริมาณสินค้าคงเหลือว่าพอให้ทำการจัดหรือไม่

- หลังจากนั้นทางแผนกคลังสินค้าจะจัดสินค้าตามใบจัดสินค้า เมื่อเรียบร้อยแล้ว แผนกบัญชี/การเงิน จะทำการออกใบแจ้งหนี้ และระบบใบแจ้งหนี้จะทำการตัดสต็อกของทางบัญชีพร้อมทั้งตั้งหนี้ในระบบบัญชีลูกหนี้ให้

16. ลักษณะของผู้ใช้

ลักษณะโดยทั่วไปของผู้ใช้ในระบบสั่งซื้อสินค้า มีดังนี้

- จบการศึกษาระดับมัธยมศึกษาตอนปลาย พร้อมทั้งได้รับการฝึกอบรมเพิ่มเติมเป็นเวลา 1-2 ปี
- มีการเลือกระบบสื่อสารที่สะดวกสบายที่สุดในการติดต่อ คือ การติดต่อสื่อสารทางโทรศัพท์
- ผู้ใช้มีความกระตือรือร้นและซื่อตรง
- ผู้ใช้ควรได้รับการฝึกอบรมในการใช้เครื่องคอมพิวเตอร์อย่างน้อย 40 ชั่วโมงต่อสัปดาห์
- ผู้ใช้ไม่ต้องมีความชำนาญในการใช้เครื่องคอมพิวเตอร์เหมือนกับนักคอมพิวเตอร์มืออาชีพ
- มีมาตรฐานในการออกแบบการต่อประสานระหว่างระบบกับผู้ดำเนินการ

17. ข้อกำหนดในการเตรียมกระบวนการในการพิจารณาซอฟต์แวร์

- ผู้ใช้ควรจะปรับขนาดของตัวอักษรก่อนที่จะพิมพ์ เพื่อเป็นการประหยัดกระดาษ
- เวลาที่ใช้ในการสำรองข้อมูลไม่ควรจะมากกว่า 6 ชั่วโมงต่อวัน

18. ความสามารถในการติดตามความต้องการ

ข้อกำหนดนี้สามารถตรวจสอบความต้องการหรือมีการทดสอบความต้องการกลับไปยังข้อกำหนดระบบของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติของบริษัท

19. ข้อกำหนด

ระบบสั่งซื้อสินค้าจะได้รับการประมวลผลภายใต้ระบบปฏิบัติการ Zebra 7 release 4 และใช้ระบบปฏิบัติการเครือข่าย GIRAFE - 12 release 3 นอกจากนี้ ระบบสั่งซื้อสินค้าจะเขียนด้วยภาษา C++ และเครื่องคอมพิวเตอร์ที่ใช้กับระบบนี้ คือ UNIGRAPH model A-3 สำหรับคีย์บอร์ดที่ใช้กับเครื่องคอมพิวเตอร์นี้ คือ คีย์บอร์ดมาตรฐาน QWERTY

20. ข้อสันนิษฐานและการขึ้นต่อกัน

- มีการตั้งข้อสันนิษฐานเกี่ยวกับลูกค้าว่าต้องมีความรู้ทางด้านกระบวนการสั่งซื้อสินค้า และการออกแบบซอฟต์แวร์ที่ดีพอ ถ้ามีการเปลี่ยนแปลงการออกแบบซอฟต์แวร์ SRS จะต้องได้รับการเปลี่ยนแปลงด้วย

- ระบบย่อยของระบบสั่งซื้อสินค้าจะดำเนินงานภายใต้ระบบปฏิบัติการ Windows 95 หรือ Windows NT และดำเนินงานบนเครื่องคอมพิวเตอร์ส่วนบุคคลรุ่นเพนเทียม

21. การแบ่งส่วนปันส่วนความต้องการ

การเพิ่มเติมความต้องการและรายละเอียดต่าง ๆ จะปรากฏบนหน้าจอคอมพิวเตอร์ ซึ่งผู้ดำเนินการกำลังใช้งานอยู่ เช่น อัตราการเปลี่ยนแปลง จำนวน ที่ตั้ง ขนาด และการแสดงคามผิดพลาด โดยผู้เชี่ยวชาญเป็นผู้กำหนดสิ่งเหล่านี้ นอกจากนี้ ผู้เชี่ยวชาญยังกำหนดมาตรฐานของหน้าจอคอมพิวเตอร์ และพิจารณาถึงจำนวนครั้งที่ผู้ดำเนินการกดคีย์บอร์ดให้น้อยที่สุด เพื่อช่วยลดความผิดพลาดต่าง ๆ ที่อาจเกิดขึ้น โดยมีระบบช่วยเหลือแบบ on-line เพื่อเป็นแนวทางให้ผู้ดำเนินการแก้ไขปัญหาต่าง ๆ ที่เกิดจากความผิดพลาดของซอฟต์แวร์ได้

ส่วนที่ 3

22. การต่อประสานภายนอก

การต่อประสานภายนอก จะรวมถึงการบำรุงรักษาซอฟต์แวร์ ผู้ดำเนินการจะรายงานและตีความหมายของเหตุการณ์ต่าง ๆ ให้กับลูกค้าทางโทรศัพท์ โดยจะมีการบำรุงรักษาซอฟต์แวร์ด้วยระบบ Philco RR model 3561 รายงานต่าง ๆ ที่เก็บบนกระดาส และเก็บลงดิสก์จะได้รับการบำรุงรักษา ซึ่งผลลัพธ์เหล่านี้ จะได้รับการควบคุมโดยบุคคลที่มีอำนาจในการจัดการข้อมูลเท่านั้น

23. ความต้องการด้านหน้าที่

หน้าที่ของระบบสั่งซื้อสินค้า ประกอบด้วย 4 ส่วน ได้แก่

23.1 หน้าที่ของการป้อนข้อมูลนำเข้าในการสั่งซื้อสินค้า

23.1.1 บทนำ ในส่วนนี้ เป็นการเก็บรวบรวมข้อมูล และแสดงรายละเอียดต่าง ๆ ของข้อมูลบนหน้าจอคอมพิวเตอร์ที่ส่งผลกระทบต่อ การสั่งซื้อของลูกค้า และการขายโดยผู้ดำเนินการ หน้าที่ในการเข้าถึงฐานข้อมูลเป็นแบบอ่านอย่างเดียว และการแสดงรายละเอียดต่าง ๆ ไปยังผู้ดำเนินการ รายละเอียดเหล่านี้ จะได้รับการถ่ายทอดไปยังลูกค้าทางโทรศัพท์ด้วย

23.1.2 ข้อมูลนำเข้า มาตรฐานของข้อมูลนำเข้าในการป้อนข้อมูลไปยังระบบสั่งซื้อสินค้า จะใช้หมายเลขประจำตัวลูกค้า (customer identification number : CID) และหมายเลขประจำตัวสินค้า (product identification number : PID) โดย CID จะเป็นอักขระตัวอักษรผสมตัวเลข 6 หลัก และ PID จะเป็นอักขระตัวอักษรผสมตัวเลข 12 หลัก

ชนิดของข้อมูลนำเข้าประเภทที่สอง คือ รหัสลูกค้าใหม่ ซึ่งลูกค้าใหม่จะไม่มี CID ผู้ดำเนินการจะป้อนข้อมูลลงในรายการเปลี่ยนแปลงของลูกค้าใหม่ เมนูจะแสดงหน้าจอของการป้อนข้อมูลนำเข้าในการสั่งซื้อสินค้า เพื่อให้กรอกชื่อลูกค้า ที่อยู่ และเบอร์โทรศัพท์ รายละเอียดเหล่านี้ จะยังคงเก็บอยู่ในหน่วยความจำจนกระทั่งทำการเปลี่ยนแปลงเสร็จสมบูรณ์ หน้าที่ของการป้อนข้อมูลนำเข้าในการสั่งซื้อสินค้าจะยอมรับชนิดของบัตรเครดิต หมายเลขบัตรเครดิต และ

วันที่หมดอายุของบัตรเครดิต ผู้ดำเนินการจะป้อนคำสั่งต่าง ๆ เพื่อตรวจสอบความถูกต้องของบัตรเครดิต เมนูจะแสดงบัญชีรายชื่อชนิดของบัตรเครดิตที่ได้รับการยอมรับ ผู้ดำเนินการสามารถกดเป็นคีย์บอร์ดเพื่อเลือกชนิดของบัตรเครดิตได้ สำหรับหมายเลขบัตรเครดิต และวันที่หมดอายุของบัตรเครดิต จะเก็บอยู่ในระเบียบของลูกค้ารายนั้น

เมื่อกดฟังก์ชันคีย์ เมนูแจ้งรายการสินค้าจะปรากฏขึ้นมา โดยมีทางเลือกดังนี้

- (1) Catalogue -- season, year (12 characters)
- (2) Flyer -- season, month, year, number (18 characters)
- (3) Home shopping -- week, month (10 characters)
- (4) Other

เมนูย่อยจะจัดหาทางเลือกต่าง ๆ ดังต่อไปนี้

- (1) PID (6 characters)
- (2) Product key words (50 characters)
- (3) Page number (6 characters)
- (4) Other

23.1.3 กระบวนการ ข้อมูลนำเข้าในการสั่งซื้อสินค้า จะใช้ PID เป็นคีย์ เพื่อที่จะดาวน์โหลดรายชื่อสินค้าที่เป็นข้อความและรูปภาพให้ปรากฏบนหน้าจอคอมพิวเตอร์ พร้อมทั้งจะดาวน์โหลดข้อมูลของลูกค้าที่สมัครโดยใช้ CID เป็นคีย์ การดาวน์โหลดข้อมูลทั้ง 2 ระเบียบ จะใช้เวลาน้อยกว่า 1 วินาที เวลาที่ใช้ในการดาวน์โหลดจะวัดจากการกดเป็นคีย์บอร์ดเพื่อป้อน PID หรือ CID ให้ไปปรากฏบนหน้าจอคอมพิวเตอร์ รายละเอียดบัตรเครดิตของลูกค้าใหม่ จะมีรูปแบบของข้อความที่เก็บรายละเอียดของบัตรเครดิต ได้แก่ type, number, expiration date, operator ID, station ID, date, และ time ข้อความต่าง ๆ จะได้รับการตรวจสอบความถูกต้องของบัตรเครดิตภายในเวลา 1 วินาที พร้อมทั้งทำการโต้ตอบหรือแสดงผลภายใน 2 วินาที ผู้ดำเนินการสามารถอ่านรายละเอียดเหล่านี้ได้อย่างชัดเจน และไม่มีความคลุมเครือเกิดขึ้น

23.1.4 ในการจัดสินค้า ใบจัดสินค้าจะอ้างอิงเอกสารใบสั่งขาย และรายการสินค้าตามใบสั่งขายซึ่งสามารถอ้างอิงได้มากกว่า 1 ใบสั่งขาย

23.1.5 เอกสารใบจัดสินค้านี้จะตัดปริมาณสินค้า ในระบบสินค้าสำเร็จรูปคงคลังตามรหัสสินค้า, ปริมาณตามวันที่ใบจัดสินค้า และพิมพ์เอกสารใบจัดสินค้า ส่งให้ระบบสินค้าสำเร็จรูปคงคลัง เพื่อเตรียมสินค้าตามที่ระบุในเอกสาร

23.1.6 รายการสินค้าที่ยกเลิกจะต้องมีในใบสั่งขาย และปริมาณที่ยกเลิกจะต้องไม่มากกว่าประมาณค้างส่งตามใบสั่งขาย

23.1.7 เอกสารใบจัดสินค้าที่ต้องการยกเลิกจะต้องยังไม่ถูกยืนยัน (Confirm) การจ่ายสินค้าและยังไม่นำไปออกเอกสารใบกำกับภาษี เอกสารใบจัดสินค้าที่ยกเลิกไปแล้วไม่สามารถนำมาจัดสินค้าได้ใหม่ และจะต้องยกเลิกรายการสินค้าทุกรายการในเลขที่ใบจัดสินค้าที่ต้องการยกเลิก ข้อมูลใบจัดสินค้าที่ยกเลิกจะต้องปรับปรุงปริมาณค่านระบบสินค้าสำเร็จรูปคงคลัง ตามรหัสสินค้า

23.1.8 ข้อมูลนำออก หน้าที่ของข้อมูลนำออกในการสั่งซื้อสินค้า จะมีรูปแบบในการส่งผ่านข้อความผ่านเครือข่ายไปยังแฟ้มข้อมูล และระบบย่อย ดังต่อไปนี้ ไปยังระบบบัญชี เพื่อบันทึกลงในทะเบียนเหล่านี้ คือ CID, PID, price, tax, shipping cost, date, time, operator ID.

23.1.9 การจัดการความผิดพลาด การป้อนข้อมูลในระบบสั่งซื้อสินค้า ถ้าข้อมูลที่ป้อนเข้าไปในระบบมีความผิดพลาดเกิดขึ้น ระบบจะโต้ตอบด้วยการแสดงข้อผิดพลาด เช่น มีอักขระมากเกินไป มีอักขระน้อยเกินไป มีอักขระที่ไม่ถูกต้อง เป็นต้น

24. ความต้องการด้านประสิทธิภาพ

เมื่อผู้ใช้ร้องขอไปยังระบบ ระบบจะโต้ตอบภายในเวลา 1 วินาที การโต้ตอบนี้ จะวัดจากการป้อนข้อมูลหรือกดแป้นคีย์บอร์ดครั้งสุดท้าย เพื่อให้ข้อมูลไปปรากฏบนหน้าจอคอมพิวเตอร์ และในการเข้าถึงฐานข้อมูลโดยทั่วไป ควรใช้เวลาภายใน 1 วินาที

25. ความต้องการด้านฐานข้อมูลเชิงตรรกะ

- การเข้าถึงฐานข้อมูลแบบอ่านอย่างเดียว และการแสดงรายละเอียดต่าง ๆ ไปยังผู้ดำเนินการ รายละเอียดเหล่านี้ จะได้รับการถ่ายทอดไปยังลูกค้าทางโทรศัพท์ด้วย
- ทะเบียนของแฟ้มข้อมูลที่มีความผิดพลาดเกิดขึ้น จะเก็บข้อมูลไว้ภายใน 6 เดือน หลังจาก 6 เดือนแล้ว ควรเก็บลงบนเทปเป็นเวลา 3 ปี หลังจากนั้น จึงจะทำลายข้อมูลนั้นทิ้ง

26. ความต้องการด้านการดำเนินการ

ระบบสั่งซื้อสินค้าจัดหาระบบช่วยเหลือให้กับผู้ใช้ เมื่อมีสถานการณ์ที่ผิดปกติในการติดต่อระหว่างผู้ใช้กับระบบเกิดขึ้น การป้อนข้อมูลในระบบสั่งซื้อสินค้า ถ้าข้อมูลที่ป้อนเข้าไปในระบบมีความผิดพลาดเกิดขึ้นระบบจะโต้ตอบด้วยการแสดงข้อความผิดพลาดเช่น มีอักขระมากเกินไป มีอักขระน้อยเกินไป มีอักขระที่ไม่ถูกต้อง เป็นต้น

27. ความต้องการด้านการทดสอบ

ผลลัพธ์ของโปรแกรม จะใช้เวลาภายใน 20 วินาทีในการประมวลผลเหตุการณ์ x 60% และจะใช้เวลาภายใน 30 วินาทีในการประมวลผลเหตุการณ์ x 99%

28. ความต้องการด้านการยอมรับ

- ระบุถึงข้อจำกัดในการตรวจรับซอฟต์แวร์

- ระบุถึงการตรวจสอบความถูกต้องในการส่งมอบซอฟต์แวร์ เพื่อดูว่าซอฟต์แวร์สามารถทำหน้าที่ได้ตรงตามความต้องการของผู้ใช้หรือลูกค้าหรือไม่

29. ความต้องการด้านเอกสาร

- เอกสารในการออกแบบ ได้แก่ ข้อกำหนดของการออกแบบซอฟต์แวร์ (Software Design Specification : SDS) โดยใช้มาตรฐาน ANSI/IEEE 1016-1987
- เอกสารซอฟต์แวร์ จะรวมถึง ข้อกำหนดในการออกแบบ ซึ่งอ้างอิงจากมาตรฐาน IEEE 830-1993 และเซตของการประมวลผลกรณีทดสอบ (test case) ที่มีผลลัพธ์เหมาะสม
- เอกสารการทดสอบซอฟต์แวร์ จะรวมถึง แผนการทดสอบระบบ (test plan) การสร้างกรณีทดสอบและผลลัพธ์จากการสร้างกรณีทดสอบ
- เอกสารการบำรุงรักษาซอฟต์แวร์ ควรจะรวมถึงการบำรุงรักษาซอฟต์แวร์ระหว่างขั้นตอนการดำเนินการ

30. ความต้องการด้านความปลอดภัย

- ระบบควรจะทำให้แน่ใจว่าข้อมูลจะไม่ได้รับความเสียหาย เมื่อมีความผิดพลาดเกิดขึ้นกับระบบ
- สถานะของเหตุการณ์ที่ผิดปกติจะบันทึกลงในรายงานประจำวันของระบบนั้น หลังจากที่มีการตัดสินใจกับเหตุการณ์ที่ผิดปกตินี้แล้ว ผู้ใช้จะสิ้นสุดการใช้งาน
- เลือกระเบียน (fields) จากรายงานที่มีความผิดพลาดเกิดขึ้น และนำระเบียนเหล่านี้พิมพ์ใส่กระดาษ (hard copy)

31. ความต้องการด้านสภาพแวดล้อม

- ในการติดตั้งซอฟต์แวร์ ควรจะมีอุณหภูมิห้องระหว่าง 18-25 องศาเซลเซียส
- ไม่ควรติดตั้งซอฟต์แวร์ใกล้ห้องเครื่องปรับอากาศ

32. ความต้องการด้านการใช้ทรัพยากรฮาร์ดแวร์คอมพิวเตอร์ให้เกิดประโยชน์สูงสุด

- เนื้อที่ว่างในฮาร์ดดิสก์ 5 เมกะไบต์ขึ้นไป
- โปรแกรมทั้งหมดจะได้รับการประมวลผลภายในโควตาของผู้ใช้ที่กำหนดไว้

33. ความต้องการด้านระบบสื่อสารคอมพิวเตอร์

- การสื่อสารข้อมูลระหว่างระบบย่อยของระบบส่งซื้อสินค้าและระบบคลังสินค้า ระบบคลังสินค้าจะส่งผ่านข้อความไปยังผู้รับ เมื่อต้นทางคือระบบคลังสินค้า
- การสื่อสารข้อมูลระหว่างระบบย่อยของระบบส่งซื้อสินค้าและระบบบัญชี ถ้าต้นทางคือระบบบัญชี จะทำการส่งข้อความเพื่อกำหนดการดำเนินการที่จะใช้ในระบบนี้

34. ปัจจัยที่มีผลต่อคุณภาพของซอฟต์แวร์

- SRS มีความต้องการกัน เช่น รูปแบบของรายงาน ถ้าในข้อกำหนดหนึ่ง อธิบายความต้องการในรูปแบบของตาราง ในข้อกำหนดอื่น ๆ ต้องอธิบายความต้องการในรูปแบบของตารางด้วย
- SRS สามารถตรวจสอบความถูกต้องของซอฟต์แวร์ เช่น ผลลัพธ์ของโปรแกรม จะใช้เวลาภายใน 20 วินาทีในการประมวลผลเหตุการณ์ x 60% และจะใช้เวลาภายใน 30 วินาทีในการประมวลผลเหตุการณ์ x 100%

35. ความต้องการด้านการฝึกอบรมบุคคลที่เกี่ยวข้อง

- ผู้พัฒนาซอฟต์แวร์ควรจะได้รับ การฝึกอบรมเกี่ยวกับภาษาที่ใช้ในการเขียนโปรแกรม ก่อนที่จะทำการพัฒนาระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติ
- ผู้พัฒนาซอฟต์แวร์จะฝึกอบรมให้กับผู้ใช้งานในเรื่องการบำรุงรักษาซอฟต์แวร์และการเปลี่ยนแปลงแก้ไขของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติ

36. ความต้องการด้านการเตรียมขั้นตอนต่าง ๆ ที่ใช้ในการพัฒนาซอฟต์แวร์

- ผู้ใช้สามารถดาวน์โหลดเวอร์ชันใหม่ของซอฟต์แวร์จากเว็บไซต์
- ผู้ใช้สามารถสอบถามปัญหาต่าง ๆ จากผู้พัฒนาซอฟต์แวร์ทางโทรศัพท์
- ผู้ใช้ควรจะสามารถสำรองข้อมูลบนเทป

37. ความต้องการด้านการบรรจุภัณฑ์ซอฟต์แวร์

- ซอฟต์แวร์นี้ จะบรรจุภัณฑ์ลงบนซีดีรอมและเทป สำหรับส่งมอบซอฟต์แวร์ให้กับลูกค้า
- การบรรจุภัณฑ์นี้ นอกเหนือจากซอฟต์แวร์ที่จะส่งมอบให้กับลูกค้าแล้ว ยังรวมถึงเอกสารต่าง ๆ ด้วย เช่น เอกสารข้อกำหนดความต้องการซอฟต์แวร์ เอกสารคู่มือผู้ใช้ซอฟต์แวร์ เอกสารการออกแบบซอฟต์แวร์ เอกสารการทดสอบซอฟต์แวร์ เอกสารการบำรุงรักษาซอฟต์แวร์ เป็นต้น

38. ข้อจำกัดในการใช้งานซอฟต์แวร์

ข้อบังคับที่ใช้ในการออกแบบซอฟต์แวร์ คือ ใช้ UNIGRAPH model A-3

39. มาตรฐานที่มีใช้ในองค์กร

การออกแบบและการพัฒนาซอฟต์แวร์ จะอ้างอิงจากมาตรฐาน ANSI/IEEE 1016-1987 และอ้างอิงจากมาตรฐานของบริษัท ถ้ามาตรฐานทั้งสองเกิดความขัดแย้งกัน จะประยุกต์ใช้มาตรฐานของบริษัทเป็นหลักในการพัฒนาซอฟต์แวร์

40. ความเชื่อถือได้

เวลาเฉลี่ยที่ซอฟต์แวร์สามารถทำงานได้ก่อนมีความผิดพลาดเกิดขึ้น จะต้องมากกว่า 5 วันขึ้นไป และภายในระยะเวลา 6 เดือน เวลาเฉลี่ยที่ซอฟต์แวร์สามารถทำงานได้ก่อนมีความผิดพลาดเกิดขึ้นโดยรวมเท่ากับ 4 สัปดาห์

41. สภาพพร้อมใช้งาน

เครื่องคอมพิวเตอร์ที่ใช้ในระบบสั่งซื้อสินค้า จะมีสภาพพร้อมใช้งาน 99.99% ซึ่งสภาพพร้อมใช้งาน ได้แก่ ระบบปฏิบัติการ โปรแกรมประยุกต์ และฮาร์ดแวร์

42. ความมั่นคง

ระบบสั่งซื้อสินค้าจะอนุญาตให้เฉพาะผู้ที่มีสิทธิ์หรือมีอำนาจในการเข้าถึงข้อมูลในระบบนี้เท่านั้น ผู้ดำเนินการ เสมิชน และผู้จัดการสามารถเข้าถึงข้อมูล เพิ่มข้อมูล และระบบย่อยอื่น ๆ ตลอดจนโปรแกรมประยุกต์ของระบบสั่งซื้อสินค้า สำหรับระบบย่อยอื่น ๆ เช่น คลังสินค้า การบัญชี เป็นต้น จะได้รับการป้องกันจากการเข้าถึงข้อมูลหรือจัดการเพิ่มข้อมูลจากระบบย่อยของระบบสั่งซื้อสินค้า

43. ความสามารถในการบำรุงรักษา

- เวลาเฉลี่ยในการแก้ไขความผิดพลาดต้องไม่เกิน 1 สัปดาห์
- เอกสารในการออกแบบ ได้แก่ ข้อกำหนดของการออกแบบซอฟต์แวร์ (Software Design Specification : SDS) โดยใช้มาตรฐาน ANSI/IEEE 1016-1987 และเอกสารการทดสอบซอฟต์แวร์จะรวมถึง แผนการทดสอบระบบ การสร้างกรณีทดสอบ และผลลัพธ์จากการสร้างกรณีทดสอบ

44.ความสามารถในการใช้ได้หลายระบบ

- ซอฟต์แวร์นี้ สามารถใช้งานได้ในสภาพแวดล้อม X และ Y
- ไม่มีส่วนใดส่วนหนึ่งของซอฟต์แวร์ที่เขียนด้วยภาษาแอสเซมบลี
- ถ้ามีการเปลี่ยนแปลงรหัสต้นทาง (Source code) น้อยกว่า 2% จะไม่ส่งผลกระทบต่อถึงการแปลโปรแกรม (compile) ใหม่อีกครั้งหนึ่งในซอฟต์แวร์นี้ เพื่อให้ดำเนินงานบนเครื่องคอมพิวเตอร์ X ได้

ส่วนที่ 4

45.สารบัญ

ส่วนที่

1	บทนำ.....1
---	------------

2	รายละเอียดโปรแกรม.....	2
3	ความต้องการที่เฉพาะเจาะจง.....	5
4	ข้อมูลสนับสนุน.....	13

46. บันทึก

หน้านี้มีไว้เพื่อบันทึกสิ่งที่ต้องการอธิบายเพิ่มเติมใน SRS

47. ภาคผนวก

- ภาคผนวก ก : ตารางของการตรวจสอบความต้องการ
- ภาคผนวก ข : สารบัญของเอกสารข้อกำหนดความต้องการซอฟต์แวร์
- ภาคผนวก ค : แผ่นพับของ SRS

48. คำชี้

- ความต้องการด้านหน้าที่, 44
- วัฏจักรของการพัฒนาซอฟต์แวร์, 3, 4
- เอกสารข้อกำหนดความต้องการซอฟต์แวร์, 1, 5, 7, 12

กรณีทดสอบ

1. ความต้องการเชิงหน้าที่

มาตรฐานของข้อมูลนำเข้าในการป้อนข้อมูลไปยังระบบสั่งซื้อสินค้า จะใช้หมายเลขประจำตัวลูกค้า (customer identification number : CID) และหมายเลขประจำตัวสินค้า (product identification number : PID)

กรณีทดสอบที่ 1

Test Case Number	Testcase01
Test Item	การป้อนหมายเลขประจำตัวลูกค้าและหมายเลขประจำตัวสินค้าไปยังระบบสั่งซื้อสินค้า
Pre-conditions	มีหมายเลขประจำตัวลูกค้าและหมายเลขประจำตัวสินค้าอยู่ในฐานข้อมูล
Post-conditions	-
Input Specifications	หมายเลขประจำตัวลูกค้า จะเป็นอักขระตัวอักษรผสมตัวเลข 6 หลัก และ หมายเลขประจำตัวสินค้า จะเป็นอักขระตัวอักษรผสมตัวเลข 12 หลัก
Expected Output Specifications	แสดงหน้าจอของการป้อนข้อมูลนำเข้าในการสั่งซื้อสินค้า
Pass/Fail Criteria	ผ่าน
Assumptions and Constraints	-
Dependencies	-
Traceability	-

กรณีทดสอบที่ 2

Test Case Number	Testcase02
Test Item	การป้อนหมายเลขประจำตัวลูกค้าและหมายเลขประจำตัวสินค้าไปยังระบบสั่งซื้อสินค้า
Pre-conditions	มีหมายเลขประจำตัวลูกค้าและหมายเลขประจำตัวสินค้าอยู่ในฐานข้อมูล
Post-conditions	-
Input Specifications	บันทึกหมายเลขประจำตัวลูกค้า จะเป็นอักขระตัวอักษรผสมตัวเลข ไม่ถึง 6 หลัก และ หมายเลขประจำตัวสินค้า จะเป็นอักขระตัวอักษรผสมตัวเลข ไม่ถึง 12 หลัก
Expected Output Specifications	แสดงข้อผิดพลาดให้ผู้ใช้งานทราบเพื่อให้พนักงานใส่ข้อมูลเข้าไปใหม่
Pass/Fail Criteria	ไม่ผ่าน
Assumptions and Constraints	-
Dependencies	-
Traceability	-

2. ความต้องการเชิงหน้าที่

ในกรณีลูกค้ารายใหม่ ซึ่งลูกค้าใหม่จะไม่มี CID ผู้ดำเนินการสามารถป้อนข้อมูลของลูกค้าใหม่ลงในรายการเปลี่ยนแปลงได้

กรณีทดสอบที่ 1

Test Case Number	Testcase03
Test Item	การป้อนข้อมูลลูกค้าใหม่
Pre-conditions	ไม่มีหมายเลขประจำตัวลูกค้า
Post-conditions	-
Input Specifications	เมนูจะแสดงหน้าจอของการป้อนข้อมูลนำเข้าในการสั่งซื้อสินค้า เพื่อให้กรอกชื่อลูกค้า ที่อยู่ และเบอร์โทรศัพท์
Expected Output Specifications	สามารถอ้างอิง CID ของลูกค้าใหม่ในโปรแกรมสั่งซื้อสินค้าได้
Pass/Fail Criteria	ผ่าน
Assumptions and Constraints	-
Dependencies	-
Traceability	-

3. ความต้องการเชิงหน้าที่

ในการจัดสินค้า ใบจัดสินค้าจะอ้างอิงเอกสารใบสั่งขาย และรายการสินค้าตามใบสั่งขาย ซึ่งสามารถอ้างอิงได้มากกว่า 1 ใบสั่งขาย

กรณีทดสอบที่ 1

Test Case Number	Testcase03
Test Item	การอ้างอิงใบสั่งขายมากกว่า 1 ใบสั่งขาย
Pre-conditions	ลูกค้า1 รายมีใบสั่งขายมากกว่า 1 ใบสั่งขาย
Post-conditions	-
Input Specifications	ใบจัดสินค้าของลูกค้ารายเดียวกัน มีการอ้างอิงใบสั่งขายมากกว่า 1 ใบ
Expected Output Specifications	ใบจัดสินค้าสามารถรวมรายการสินค้าจากใบสั่งขายหลายใบได้
Pass/Fail Criteria	ผ่าน
Assumptions and Constraints	-
Dependencies	-
Traceability	-

4. ความต้องการเชิงหน้าที่

เอกสารใบจัดสินค้านี้จะตัดปริมาณสินค้า ในระบบสินค้าสำเร็จรูปคงคลัง ตามรหัสสินค้า, ปริมาณตามวันที่ใบจัดสินค้า และพิมพ์เอกสารใบจัดสินค้า ส่งให้ระบบสินค้าสำเร็จรูปคงคลัง เพื่อเตรียมสินค้าตามที่ระบุในเอกสาร

กรณีทดสอบที่ 1

Test Case Number	Testcase04
Test Item	การตัดปริมาณสินค้าในระบบสินค้าสำเร็จรูปคงคลัง
Pre-conditions	ตรวจสอบปริมาณสินค้าก่อนการทำใบจัดสินค้า
Post-conditions	ตรวจสอบปริมาณสินค้าหลังการทำใบจัดสินค้า
Input Specifications	บันทึกใบจัดสินค้าของลูกค้ารายเดียวกัน
Expected Output Specifications	ปริมาณสินค้าในระบบสินค้าสำเร็จรูปคงคลังถูกปรับยอดลดลงตามปริมาณสินค้าที่ระบุในใบจัดสินค้า
Pass/Fail Criteria	ผ่าน
Assumptions and Constraints	-
Dependencies	-
Traceability	-

5. ความต้องการเชิงหน้าที่

รายการสินค้าที่ยกเลิกต้องมีในใบสั่งขาย และปริมาณที่ยกเลิกจะต้องไม่มากกว่าประมาณ
ค้างส่งตามใบสั่งขาย

กรณีทดสอบที่ 1

Test Case Number	Testcase05
Test Item	การยกเลิกรายการสินค้าในใบสั่งขาย
Pre-conditions	ตรวจสอบยอดค้างส่งรายการสินค้าตามใบสั่งขายที่จะยกเลิก
Post-conditions	ใบสั่งขายที่ถูกยกเลิกไม่สามารถนำไปใช้ใหม่ได้
Input Specifications	ระบุจำนวนรายการสินค้าที่ต้องการยกเลิกตามยอดค้างส่งในใบสั่งขาย
Expected Output Specifications	ใบสั่งขายถูกยกเลิกและปริมาณสินค้าในระบบสินค้าสำเร็จรูปคง คลังถูกปรับปรุงเพิ่มขึ้นตามจำนวนที่ถูกยกเลิก
Pass/Fail Criteria	ผ่าน
Assumptions and Constraints	-
Dependencies	-
Traceability	-

6. ความต้องการเชิงหน้าที่

เอกสารใบจัดสินค้าที่ต้องการยกเลิกจะต้องยังไม่ถูกยืนยัน (Confirm) การจ่ายสินค้าและยังไม่นำไปออกเอกสารใบกำกับภาษี เอกสารใบจัดสินค้าที่ยกเลิกไปแล้วไม่สามารถนำมาจัดสินค้าได้ใหม่ และจะต้องยกเลิกรายการสินค้าทุกรายการในเลขที่ใบจัดสินค้าที่ต้องการยกเลิก ข้อมูลใบจัดสินค้าที่ยกเลิกจะต้องปรับปรุงปริมาณคิในระบบสินค้าสำเร็จรูปคงคลัง ตามรหัสสินค้า

กรณีทดสอบที่ 1

Test Case Number	Testcase06
Test Item	การยกเลิกรายการสินค้าในใบจัดสินค้า
Pre-conditions	ใบจัดสินค้าที่ถูกบันทึกและยังไม่ถูกนำไปออกใบกำกับภาษี
Post-conditions	ไม่สามารถนำใบจัดสินค้าที่ถูกยกเลิกไปใช้ใหม่ได้
Input Specifications	กดปุ่มยืนยันการยกเลิกเอกสารใบจัดสินค้า
Expected Output Specifications	ยกเลิกใบจัดสินค้า รายการสินค้าทุกรายการในใบจัดสินค้าถูกยกเลิกทั้งหมด และปริมาณสินค้าในระบบสินค้าสำเร็จรูปคงคลังถูกปรับปรุงเพิ่มขึ้นตามปริมาณที่ถูกยกเลิก
Pass/Fail Criteria	ผ่าน
Assumptions and Constraints	-
Dependencies	-
Traceability	-

7. ความต้องการเชิงหน้าที่

การจัดการความผิดพลาด การป้อนข้อมูลในระบบสั่งซื้อสินค้า ถ้าข้อมูลที่ป้อนเข้าไปในระบบมีความผิดพลาดเกิดขึ้น ระบบจะโต้ตอบด้วยการแสดงข้อผิดพลาด เช่น มีอักขระมากเกินไป มีอักขระน้อยเกินไป มีอักขระที่ไม่ถูกต้อง เป็นต้น

กรณีทดสอบที่1

Test Case Number	Testcase07
Test Item	การแสดงผลผิดพลาดในระบบ
Pre-conditions	-
Post-conditions	-
Input Specifications	บันทึกข้อมูลที่มีอักขระมากเกินไปและไม่ถูกต้อง
Expected Output Specifications	แสดงผลผิดพลาดที่เกิดขึ้นให้กับผู้ใช้งานทราบและกลับไปยังจุดเดิมก่อนการเกิดข้อผิดพลาด
Pass/Fail Criteria	ผ่าน
Assumptions and Constraints	-
Dependencies	-
Traceability	-













ภาคผนวก ค

คู่มือการใช้งานเครื่องมือสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ







● มาตรฐานปุ่มคำสั่ง (Standard Toolbar)

คู่มือการใช้งานระบบมี Toolbar ต่าง ๆ ที่เพิ่มความสะดวกในการทำงาน ซึ่งแสดงอยู่บนจอภาพ Toolbars แต่ละปุ่มจะมีสัญลักษณ์ที่สื่อให้เข้าใจได้ง่าย และเพื่อให้เข้าใจตรงกันสำหรับการใช้งานนั้นได้ โดยแสดงภาพ และความหมาย ที่เป็นมาตรฐาน (Standard) ของระบบนี้ เพื่อเพิ่มความเข้าใจในการปฏิบัติงานบันทึกข้อมูล หรือการใช้งานอื่น ๆ ชัดเจนยิ่งขึ้น



TOOLBARS	OPTION	คำอธิบาย
	เพิ่ม	การบันทึกข้อมูลเพิ่ม
	ลบ	การลบข้อมูล
	บันทึก	การเก็บบันทึกข้อมูล
	ค้นหา	การสอบถาม/ค้นหาข้อมูลเฉพาะกลุ่ม
	ดึงข้อมูล	ดึงข้อมูลตั้งแต่ต้นเพิ่มมาทั้งหมด
	กลับไปก่อนหน้า	ทำการย้อนข้อมูลกลับไปก่อนหน้าที่จะทำการแก้ไข
	แรกสุด	การยกเลิกการสอบถาม/ค้นหาข้อมูลเฉพาะกลุ่ม
	ก่อนหน้า	เพื่อย้อนไปยังข้อมูล Record หรือหน้าที่ผ่านมา
	ถัดไป	เพื่อไปยังข้อมูล Record หรือ หน้าต่อไป
	ท้ายสุด	ไปยังข้อมูลรายการล่างสุด (รายการสุดท้าย)
	พิมพ์	พิมพ์ข้อมูลออกทางเครื่องพิมพ์
	ออก	จบการทำงานเพื่อออกไปสู่เมนูหลัก

● วิธีการใช้งานมาตรฐานปุ่มคำสั่ง

- **การเพิ่มข้อมูล** หากมีการเพิ่มข้อมูลให้ Click Mouse ไปที่ปุ่ม  มีแถบสว่างพร้อมจะบันทึกข้อมูล
- **การค้นหาข้อมูล** ให้ Click Mouse ที่ปุ่ม  จากนั้นเลือกข้อมูลที่ต้องการค้นหา และพิมพ์ค่าที่ต้องการค้นหาลงไป แล้ว Click Mouse ที่ปุ่ม  โปรแกรมก็จะทำการดึงข้อมูลที่ต้องการมาให้
- **การแก้ไขข้อมูล** ให้ค้นหาข้อมูลที่ต้องการแก้ไขตามขั้นตอนการค้นหาข้อมูล จากนั้นให้เลื่อนแถบสว่างไปยังข้อมูลที่ต้องการแก้ไข
- **การบันทึกข้อมูล** หลังจากทำการป้อนข้อมูลที่ต้องการครบแล้วให้ Click Mouse ที่ปุ่ม  เพื่อทำการบันทึกข้อมูล
- **การลบข้อมูล** ให้ค้นหาข้อมูลที่ต้องการลบตามขั้นตอนการค้นหาข้อมูล เมื่อค้นหาพบให้ Click Mouse ที่ปุ่ม  แล้ว Click Mouse ที่ปุ่ม  เพื่อทำการยืนยันการลบข้อมูล

● ขั้นตอนการทำงานของระบบ

การทดสอบเริ่มจากผู้ทดสอบต้องพิมพ์เอกสารตัวแบบในส่วนของข้อกำหนดความต้องการซอฟต์แวร์ เพื่อนำไปใช้ในการเก็บความต้องการ แล้วนำความต้องการที่ได้มาบันทึกลงในระบบโดยใช้โปรแกรมบันทึกข้อกำหนดความต้องการซอฟต์แวร์ โดยก่อนการบันทึกข้อกำหนดความต้องการซอฟต์แวร์ ผู้ทดสอบต้องบันทึกข้อมูลในส่วนของเพิ่มข้อมูลหลักให้เรียบร้อยก่อน ซึ่งประกอบไปด้วยโปรแกรม บันทึกรายละเอียดพนักงาน, บันทึกรายละเอียดลูกค้า, บันทึกรายละเอียดRelease และ บันทึกรายละเอียดโครงการ หลังจากนั้นผู้ทดสอบจึงทำการสร้างกรณีทดสอบ โดยกดปุ่ม เงื่อนไข (Condition) ซึ่งจะปรากฏอยู่ในรายการความต้องการด้านหน้าที่สำหรับการสร้างกรณีทดสอบ ผู้ทดสอบจะต้องทำการวิเคราะห์ความต้องการ และอาศัยหลักการการออกแบบกรณีทดสอบแบบแบล็กบ็อกซ์ โดยใช้เทคนิคการสร้าง Thread Testing ซึ่งเป็นลักษณะของการทดสอบตามงานที่ทำ ในลักษณะของการใช้งานระบบ เพื่อตรวจสอบความต้องการทางด้านธุรกิจ เมื่อผู้ทดสอบทำการวิเคราะห์เรียบร้อยแล้ว จึงนำผลการวิเคราะห์มาบันทึกลงในโปรแกรมบันทึกข้อมูลกรณีทดสอบ

เมื่อผู้ทดสอบทำการบันทึกผลการทดสอบเข้าสู่ระบบเรียบร้อยแล้ว สามารถพิมพ์รายงานกรณีทดสอบต่าง ๆ ได้ โดยรายละเอียดในการสร้างกรณีทดสอบ มีดังนี้

- พิมพ์ตัวต้นแบบในส่วนของข้อกำหนดความต้องการซอฟต์แวร์
- บันทึกข้อมูลในส่วนของเพิ่มข้อมูลหลัก
- บันทึกข้อกำหนดความต้องการซอฟต์แวร์
- บันทึกกรณีทดสอบของแต่ละความต้องการด้านหน้าที่
- พิมพ์เอกสารกรณีทดสอบเพื่อนำไปทำการทดสอบ

เมนูของเครื่องมือสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ

เมนูหลัก	เมนูย่อย
1. กำหนดเพิ่มข้อมูลหลัก	1.1 กำหนดรายละเอียดพนักงาน (Employee File)
	1.2 กำหนดรายละเอียดลูกค้า (Customer File)
	1.3 กำหนดรายละเอียด Release (Release File)
	1.4 กำหนดรายละเอียดโครงการ (Project File)
2. รายการประจำวัน	2.1 บันทึกข้อกำหนดความต้องการซอฟต์แวร์ (SRS File)
3. สอบถาม	3.1 สอบถามกรณีทดสอบ (Testcase Query)
4. รายงาน	4.1 พิมพ์ต้นแบบของเอกสารความต้องการซอฟต์แวร์ (SRS Template)
	4.2 พิมพ์เอกสารกรณีทดสอบ (Testcase Report)

[illegible]


ชื่อโปรแกรม กำหนดรายละเอียดพนักงาน (Employee File)



วัตถุประสงค์ เพื่อใช้เก็บรายละเอียดของพนักงาน

อธิบายความหมาย

คำศัพท์	ความหมาย
Employee No.	ระบุรหัสพนักงาน
Employee Name	ระบุรายละเอียดพนักงาน
Update By	แสดงผู้ปรับปรุง
Update Date	แสดงวันที่ปรับปรุง

วิธีการใช้งาน

1. ระบursorพนักงาน, รายละเอียดพนักงาน ที่ต้องการบันทึก
2. บันทึกข้อมูลโดย Click Mouse ที่ปุ่ม 
3. กรณีที่ต้องการทำการรายการเพิ่มให้ทำตามขั้นตอนที่ 1 จนถึงการบันทึกข้อมูล

4. กรณีต้องการลบข้อมูล ให้เลื่อนแถบสว่างไปที่ข้อมูลที่ต้องการแล้ว Click Mouse ที่ปุ่ม  จากนั้น Click Mouse ที่ปุ่ม 

[illegible]

ชื่อโปรแกรม กำหนดรายละเอียดลูกค้า (Customer File)



วัตถุประสงค์ เพื่อใช้เก็บรายละเอียดของลูกค้า

อธิบายความหมาย

คำศัพท์	ความหมาย
Customer Code	ระบุรหัสลูกค้า
Customer Name	ระบุรายละเอียดลูกค้า
Update By	แสดงผู้ปรับปรุง
Update Date	แสดงวันที่ปรับปรุง

วิธีการใช้งาน

1. ระบุนรหัสลูกค้า, รายละเอียดลูกค้า ที่ต้องการบันทึก
2. บันทึกข้อมูลโดย Click Mouse ที่ปุ่ม 
3. กรณีที่ต้องการทำรายการเพิ่มให้ทำตามขั้นตอนที่ 1 จนถึงการบันทึกข้อมูล

4. กรณีต้องการลบข้อมูล ให้เลื่อนแถบสว่างไปที่ข้อมูลที่ต้องการแล้ว Click Mouse ที่ปุ่ม  จากนั้น Click Mouse ที่ปุ่ม 

3. กรณีที่ต้องการทำการเพิ่มให้ทำตามขั้นตอนที่ 1 จนถึงการบันทึกข้อมูล
4. กรณีต้องการลบข้อมูล ให้เลื่อนแถบสว่างไปที่ข้อมูลที่ต้องการแล้ว Click Mouse ที่ปุ่ม  จากนั้น Click Mouse ที่ปุ่ม 

Program Test Co., Ltd. \ Suranaree University of Technology \ Project File

Program ID PROJECT User ID SUT

Customer Code 01 ABC Co., Ltd.

Release Code 1 Release 1

Project Code	Project Name
01	Project 1
02	Project 2
03	Project 3
04	Project 4
05	Project 5
06	Project 6
07	Project 7
08	Project 8
09	Project 9
10	Project 10
11	Project 11
12	Project 12
13	Project 13
14	Project 14
15	Project 15

Update By SUT Update Date 01/03/2006

Record: 1/?



ชื่อโปรแกรม กำหนดรายละเอียดโครงการ (Project File)

วัตถุประสงค์ เพื่อใช้เก็บรายละเอียดโครงการ

อธิบายความหมาย

คำศัพท์	ความหมาย
Customer Code	ระบุรหัสลูกค้า
Release Code	ระบุรหัสrelease
Project Code	ระบุรหัสโครงการ
Project Name	ระบุรายละเอียดโครงการ
Update By	แสดงผู้ปรับปรุง
Update Date	แสดงวันที่ปรับปรุง

วิธีการใช้งาน

1. ระบุนรหัสลูกค้า, ระบุนรหัสrelease , ระบุนรหัสโครงการ และรายละเอียด Release ที่ต้องการบันทึก
2. บันทึกข้อมูล โดย Click Mouse ที่ปุ่ม 
3. กรณีที่ต้องการทำการเพิ่มให้ทำตามขั้นตอนที่ 1 จนถึงการบันทึกข้อมูล
4. กรณีต้องการลบข้อมูล ให้เลื่อนแถบสว่างไปที่ข้อมูลที่ต้องการแล้ว Click Mouse ที่ปุ่ม  จากนั้น Click Mouse ที่ปุ่ม 

Program Test Co., Ltd. \ Suranaree University of Technology \ Software Requirements Specification File

Program ID TESTXXX User ID SUT

Customer Code 01 ABC Co., Ltd.
Release Code 1 Release 1
Project Code 01 Project 1
Srs Code SRS20060301 Software Requirements Specification 01 Ref No REF20060301
Prepare By 01 sasasasas Prepare Date 02/03/2006

SECTION01 SECTION02(2/1) SECTION02(2/2) SECTION03(3/1) SECTION03(3/2) SECTION03(3/3) SECTION04

Train Related Req

Verification Req

Functional Requirement

F20060301	Functional Requirements 01	Condition
F20060302	Functional Requirements 02	Condition
		Condition
		Condition
		Condition

ชื่อโปรแกรม บันทึกข้อกำหนดความต้องการซอฟต์แวร์ (SRS File)

วัตถุประสงค์ เพื่อใช้เก็บรายละเอียดของข้อกำหนดความต้องการซอฟต์แวร์และกรณีทดสอบที่เกี่ยวข้องกัน

อธิบายความหมาย

คำศัพท์	ความหมาย
Customer Code	รหัสลูกค้า
Release Code	รหัส release
Project Code	รหัสโครงการ
SRS Code	รหัสข้อกำหนดความต้องการซอฟต์แวร์
	รหัสรายละเอียดข้อกำหนดความต้องการซอฟต์แวร์
Ref No.	เลขที่เอกสารที่มีการอ้างอิง
Prepare By	แสดงผู้ปรับปรุง
Prepare Date	แสดงวันที่ปรับปรุง

Section 1

คำศัพท์	ความหมาย
Abbreviation Code	รหัสคำย่อ
	รายละเอียดคำย่อ
Acronyms Code	รหัสคำย่อ
	รายละเอียดคำย่อ
Definition Code	รหัสคำจำกัดความ
	รายละเอียดคำจำกัดความ
Overview Code	รหัสเนื้อหาโดยรวม
	รายละเอียดเนื้อหาโดยรวม
Purpose Code	รหัสจุดประสงค์
	รายละเอียดจุดประสงค์
Reference Code	รหัสเอกสารอ้างอิง
	รายละเอียดเอกสารอ้างอิง
Scope Code	รหัสขอบเขต
	รายละเอียดขอบเขต

Section 2

คำศัพท์	ความหมาย
Apport Require Code	รหัสการแบ่งสันปันส่วนความต้องการ
	รายละเอียดการแบ่งสันปันส่วนความต้องการ
Assump_Depend_code	รหัสข้อสันนิษฐานและการขึ้นต่อกัน
	รายละเอียดข้อสันนิษฐานและการขึ้นต่อกัน
Community Code	รหัสการต่อประสานกับระบบสื่อสาร
	รายละเอียดการต่อประสานกับระบบสื่อสาร
Constraint Code	รหัสข้อจำกัด
	รายละเอียดข้อจำกัด
Hardware Code	รหัสการต่อประสานกับฮาร์ดแวร์
	รายละเอียดการต่อประสานกับฮาร์ดแวร์
Memory Code	รหัสข้อจำกัดของหน่วยความจำ

Section 2 (ต่อ)

คำศัพท์	ความหมาย
	รายละเอียดข้อกำหนดของหน่วยความจำ
Operation code	รหัสการดำเนินการ
	รายละเอียดการดำเนินการ
Production Function Code	รหัสหน้าที่ของซอฟต์แวร์
	รายละเอียดหน้าที่ของซอฟต์แวร์
Qua Prov Code	รหัสข้อกำหนดในการเตรียมกระบวนการในการพิจารณาซอฟต์แวร์
	รายละเอียดข้อกำหนดการเตรียมกระบวนการในการพิจารณาซอฟต์แวร์
Requirement Trace Code	รหัสความสามารถในการติดตามความต้องการ
	รายละเอียดความสามารถในการติดตามความต้องการ
Site Adap Req Code	รหัสความต้องการด้านการติดตั้งซอฟต์แวร์
	รายละเอียดความต้องการด้านการติดตั้งซอฟต์แวร์
Software Code	รหัสการต่อประสานกับซอฟต์แวร์
	รายละเอียดการต่อประสานกับซอฟต์แวร์
User Interface Code	รหัสการต่อประสานกับผู้ใช้
	รายละเอียดการต่อประสานกับผู้ใช้
Acceptance Require Code	รหัสความต้องการด้านการยอมรับ
	รายละเอียดความต้องการด้านการยอมรับ

Section 3

คำศัพท์	ความหมาย
Availability Code	รหัสสภาพพร้อมใช้งาน
	รายละเอียดสภาพพร้อมใช้งาน
Comp Hard Resource Code	รหัสความต้องการด้านการใช้ทรัพยากรฮาร์ดแวร์คอมพิวเตอร์
	รายละเอียดความต้องการด้านการใช้ทรัพยากรฮาร์ดแวร์คอมพิวเตอร์
Doc Req Code	รหัสความต้องการด้านเอกสาร
	รายละเอียดความต้องการด้านเอกสาร

Section 3 (ต่อ)

Ext Interf Code	รหัสการต่อประสานภายนอก
	รายละเอียดการต่อประสานภายนอก
Func Req Code	รหัสความต้องการด้านหน้าที่
	รายละเอียดความต้องการด้านหน้าที่
Imp Cons Code	รหัสข้อจำกัดในการใช้งานซอฟต์แวร์
	รายละเอียดข้อจำกัดในการใช้งานซอฟต์แวร์
Logic Dat Req Code	รหัสความต้องการด้านฐานข้อมูลเชิงตรรกะ
	รายละเอียดความต้องการด้านฐานข้อมูลเชิงตรรกะ
Log Related Req Code	รหัสความต้องการด้านการเตรียมขั้นตอนต่าง ๆ
	รายละเอียดความต้องการด้านการเตรียมขั้นตอนต่าง ๆ
Maintainability Code	รหัสความสามารถในการบำรุงรักษา
	รายละเอียดความสามารถในการบำรุงรักษา
Operate Req Code	รหัสความต้องการด้านการดำเนินการ
	รายละเอียดความต้องการด้านการดำเนินการ
Package Code	รหัสความต้องการด้านการบรรจุภัณฑ์
	รายละเอียดความต้องการด้านการบรรจุภัณฑ์
Perform Req Code	รหัสความต้องการด้านประสิทธิภาพ
	รายละเอียดความต้องการด้านประสิทธิภาพ
Portability Code	รหัสความสามารถในการใช้ได้หลายระบบ
	รายละเอียดความสามารถในการใช้ได้หลายระบบ
Reliability Code	รหัสความเชื่อถือได้
	รายละเอียดความเชื่อถือได้
Safety Req Code	รหัสความต้องการด้านความปลอดภัย
	รายละเอียดความต้องการด้านความปลอดภัย
Security Code	รหัสความมั่นคง
	รายละเอียดความมั่นคง
Soft Qua Fac Code	รหัสปัจจัยที่มีผลต่อคุณภาพ
	รายละเอียดปัจจัยที่มีผลต่อคุณภาพ
Standard Comp Code	รหัสมาตรฐานที่มีใช้ในองค์กร
	รายละเอียดมาตรฐานที่มีใช้ในองค์กร





Section 3 (ต่อ)

Func Req Code	รหัสความต้องการด้านหน้าที่
Testcase no	รหัสกรณีทดสอบ
Test case item	รายละเอียดของกรณีทดสอบ
Pre - Condition	เงื่อนไขก่อนใช้กรณีทดสอบ
Post - Condition	เงื่อนไขหลังใช้กรณีทดสอบ
Expected Output	ผลลัพธ์ที่ต้องการ
Pass / fail	สถานะผ่าน/ไม่ผ่าน
Assumption & Constraints	ข้อสมมุติฐานและข้อจำกัด
Dependencies	การขึ้นตรงต่อกัน
Input No.	รหัสข้อมูลเข้า
Input Specification	รายละเอียดข้อมูลเข้า
Train Related Req Code	รหัสความต้องการด้านการฝึกอบรมบุคคลที่เกี่ยวข้อง
	รายละเอียดความต้องการด้านการฝึกอบรมบุคคลที่เกี่ยวข้อง
Verification Req Code	รหัสความต้องการด้านการทวนสอบ
	รายละเอียดความต้องการด้านการทวนสอบ

Section 4

คำศัพท์	ความหมาย
Appendix Code	รหัสภาคผนวก
	รายละเอียดภาคผนวก
Index Code	รหัสดัชนี
	รายละเอียดดัชนี
Note Code	รหัสบันทึก
	รายละเอียดบันทึก
Table Content Code	รหัสสารบัญ
	รายละเอียดสารบัญ

วิธีการใช้งาน

1. ระบุรหัสลูกค้า, ระบุรหัสrelease , ระบุรหัสโครงการ และระบุรหัสข้อกำหนดความต้องการ ที่ต้องการบันทึก
2. ระบุข้อมูลที่ได้มาจาก ตัวต้นแบบข้อกำหนดความต้องการลงไปตามรูปแบบที่จัดให้
3. ถ้าต้องการบันทึกกรณีทดสอบ สามารถกำหนดได้โดยกดปุ่ม  Condition
ในส่วนของ Section ที่3/3 ซึ่งจะปรากฏอยู่ในหัวข้อของ Functional Requirement
4. ทำการบันทึกกรณีทดสอบตามเงื่อนไขที่กำหนดไว้
5. บันทึกข้อมูลโดย Click Mouse ที่ปุ่ม 
6. กรณีที่ต้องการทำการเพิ่มให้ทำตามขั้นตอนที่ 1 จนถึงการบันทึกข้อมูล
7. กรณีต้องการลบข้อมูล ให้เลื่อนแถบสว่างไปที่ข้อมูลที่ต้องการแล้ว Click Mouse ที่ปุ่ม  จากนั้น Click Mouse ที่ปุ่ม 

Program Test Co., Ltd. \ Suranaree University of Technology \ Project File

Program ID **ESTCASE_QUER** User ID **SUT**

Customer Code **01** ABC Co., Ltd.

Release Code **1** Release 1

Project Code **01** Project 1

SRS Code **SRS20060301** Software Requirements Specification 01

Functional Requirement **F20060301** Functional Requirements 01

Testcase No **TC20060301** Brief description of the test items exercised by this test case

Pre - Condition Conditions must be established before the test

Post - Condition Conditions must be established after the test

Input No Input Specification

INT01 Bullet items to describe each input required for this test case 01

INT02 Bullet items to describe each input required for this test case 02

Expected output Bullet items to describe each expected output for this test case

Pass / Fail ☒ Pass ☐ Fail

Assumption & Constraints Bullet items

Dependencies List all the test cases depend on or are depend by this test case

Record: 1/?

ชื่อโปรแกรม

สอบถามกรณีทดสอบ (Testcase Query)

วัตถุประสงค์



เพื่อใช้ตรวจสอบรายละเอียดเกี่ยวกับกรณีทดสอบที่ทำการบันทึกลงในฐานข้อมูลแล้ว

อธิบายความหมาย

คำศัพท์	ความหมาย
Customer Code	แสดงรหัสลูกค้า
Release Code	แสดงรหัส release
Project Code	แสดงรหัสโครงการ
SRS Code	แสดงรหัสข้อกำหนดความต้องการ
	แสดงรายละเอียดข้อกำหนดความต้องการ
Functional Requirement	แสดงรหัสความต้องการเชิงหน้าที่
	แสดงรายละเอียดความต้องการเชิงหน้าที่
Testcase No.	รหัสกรณีทดสอบ
Pre - Condition	เงื่อนไขก่อนใช้กรณีทดสอบ

คำศัพท์	ความหมาย
Post - Condition	เงื่อนไขหลังใช้กรณีทดสอบ
Input No.	รหัสข้อมูลเข้า
Input Specification	รายละเอียดข้อมูลเข้า
Expected Output	ผลลัพธ์ที่ต้องการ
Pass / fail	สถานะผ่าน/ไม่ผ่าน
Assumption & Constraints	ข้อสมมุติฐานและข้อจำกัด
Dependencies	การขึ้นตรงต่อกัน

วิธีการใช้งาน

1. ถ้าต้องการระบุเงื่อนไขในการสอบถามให้กดปุ่ม  และระบุเงื่อนไขในช่องสีขาว แล้วกดปุ่ม  เพื่อแสดงข้อมูลตามเงื่อนไขที่ระบุ

Program Test Co., Ltd. \ Suranaree University of Technology \ Testcase Report

Program ID SRS User ID SUT

FROM TO

Template for Software Requirements Specification

Description

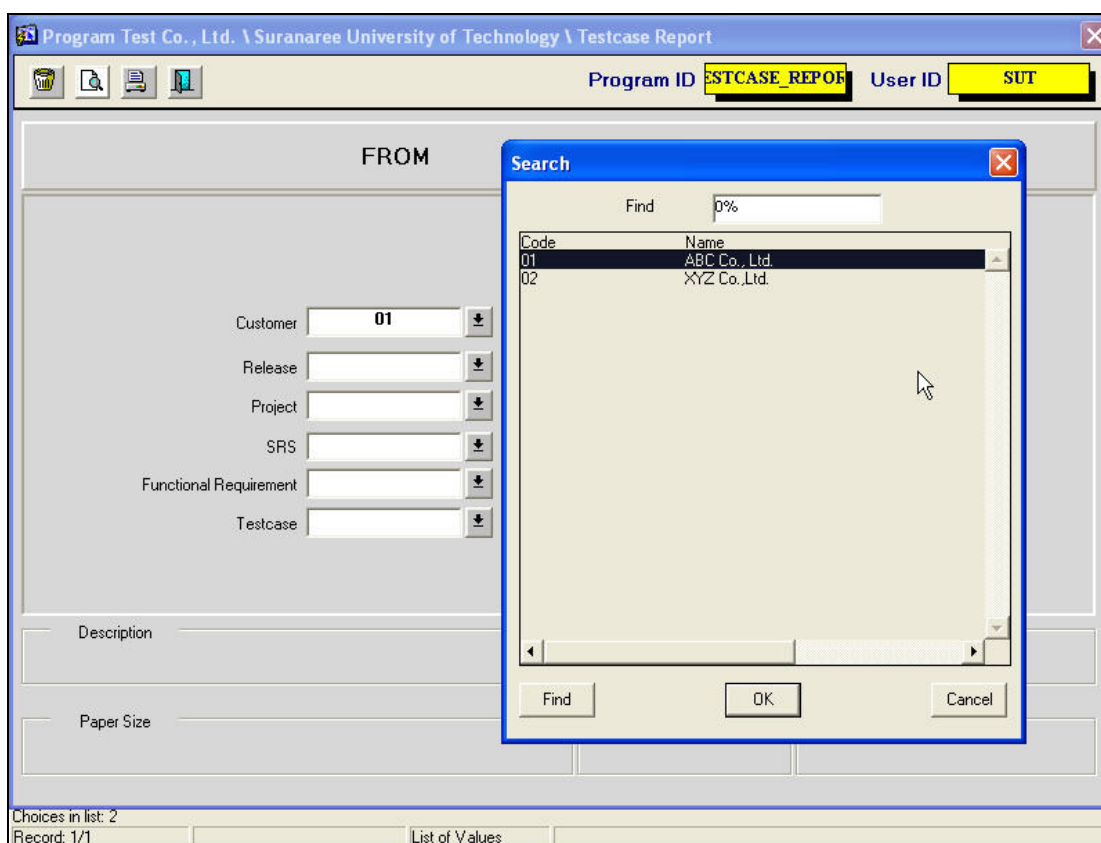
Paper Size Orientation Mode

ชื่อโปรแกรม พิมพ์ตัวต้นแบบเอกสารข้อกำหนดความต้องการซอฟต์แวร์

วัตถุประสงค์ เพื่อเป็นต้นแบบและเป็นแนวทางในการจัดเก็บความต้องการให้ตรงรูปแบบและครอบคลุมความต้องการมากที่สุด

วิธีการใช้งาน

1. Click Mouse ที่ปุ่ม  เพื่อพิมพ์รายงานทางเครื่องพิมพ์



ชื่อโปรแกรม

พิมพ์เอกสารกรณีทดสอบ (Test case Report)

วัตถุประสงค์

เพื่อนำเอกสารกรณีทดสอบที่ได้บันทึกลงในฐานข้อมูลไปทำการทดสอบ

วิธีการใช้งาน

1. กำหนดช่วงข้อมูลที่ต้องการ (เริ่มต้น - สิ้นสุด)

- รหัสลูกค้า : ระบรหัสลูกค้า
- รหัสrelease : ระบรหัสrelease
- รหัสโครงการ : ระบรหัสโครงการ
- รหัส SRS : ระบรหัส SRS
- รหัสความต้องการเชิงหน้าที่ : ระบรหัสความต้องการเชิงหน้าที่
- รหัสกรณีทดสอบ : ระบรหัสกรณีทดสอบ

2. Click Mouse ที่ปุ่ม เพื่อพิมพ์รายงานทางเครื่องพิมพ์

ประวัติผู้เขียน

นายณริศ มิ่งโมรา เกิดเมื่อวันที่ 24 เมษายน พ.ศ. 2516 เกิดที่ จังหวัดนครราชสีมา เริ่มศึกษา
ชั้นประถมศึกษาปีที่ 1-6 ที่โรงเรียนมารีย์วิทยา ชั้นมัธยมศึกษาปีที่ 1-6 ที่โรงเรียนราชสีมาวิทยาลัย
และสำเร็จการศึกษาระดับปริญญาตรี สาขาบริหารธุรกิจบัณฑิต (คอมพิวเตอร์ธุรกิจ) จาก
มหาวิทยาลัยวงษ์ชวลิตกุล จังหวัดนครราชสีมา เมื่อ พ.ศ.2538 ภายหลังสำเร็จการศึกษาได้เข้าทำงาน
กับบริษัทซอฟต์แวร์ 1999 จำกัด เป็นเวลา 9 ปี

ปี พ.ศ. 2547 เข้าศึกษาต่อในระดับปริญญาโท สาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชา
วิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

ผลงานวิจัย : ได้เสนอบทความเข้าร่วมประชุมวิชาการวิทยาศาสตร์และเทคโนโลยีแห่ง
ประเทศไทย ครั้งที่ 32 เรื่องการพัฒนาเครื่องมือสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ
ระหว่างวันที่ 10 – 12 ตุลาคม พ.ศ. 2549 ณ ศูนย์ประชุมแห่งชาติสิริกิติ์ กรุงเทพมหานคร